EL2805 Reinforcement Learning

Exercise Session 2

November 5, 2019

Department of Automatic Control
School of Electrical Engineering
KTH Royal Institute of Technology

2 Exercises

Some of these exercises have been inspired by, or taken from, [1–3]. If you want to solve more exercises, see any of those books.

2.1

Model the following problems using a Markov Decision Process (MDP). In other words, establish the time horizon, state and action spaces, the transition probabilities, and the rewards.

(a) You observe a fair coin being tossed T times. You may stop observing at any time, and when you do you receive as a reward the average number of heads observed. Thus, if the first toss is heads, you should certainly stop since your payoff is one and you can never receive a higher payoff than that. Your problem is to identify a stopping rule maximizing the average reward.

(b) A dictator is aiming at sequentially passing a series of N laws $l_1, l_2, ..., l_N$ that would allow him to accumulate wealth. His starting wealth is w_0 and each law will double his wealth, however, each law carries a probability p_r that the people revolt and a probability p_p that the parliament rejects the law. If the parliament rejects a law, the dictator does not gain any wealth (the law cannot be re-introduced). However if the people revolt, the dictator loses all his fortune and is overthrown. The two events (revolt and rejection) are independent. The goal is to retire having accumulated the largest fortune.

(c) You get to observe a fair 6-sided dice being rolled $T > 21$ times and generating outcomes $X(i)$, $i \in \{1, ..., T\}$. You may stop observing at any time before T and when you do, denoting by s the sum of all observations up to this point, you receive reward $s - 21$ if $s \leq 21$ or -10 otherwise.

2.2

You have to sell your apartment within N days, and want to maximize your profit. Every evening, you receive an offer w_t, which you have to either accept or reject the next day. (A rejected offer cannot be called back.) Assume that all offers are multiples of 10 000 SEK, and that they are i.i.d., positive and bounded. Their distribution is known to you from previous sales in your neighbourhood. Once you sell your apartment, you receive a fixed daily interest rate $\rho > 0$ on the money that land in your bank account. Model this as an MDP.

2.3

One of the most well-known applications of Markov chains in the real world is the infamous PageRank algorithm (proposed by Larry Page and Sergey Brin in [4]), which sparked the creation of the Google search engine in the late '90s. In their vision, users followed a random walk through the Internet, moving from one page to another via hyperlinks, uniformly at random. However, users are considered to interrupt this walk and jump to a different web page with probability α (Brin and Page consider $\alpha = 0.15$). In light of this model of user behavior, the probability of a page being relevant is deemed proportional to the fraction of times users visit it, and thus, is given by the stationary distribution π of the Markov chain described above.

Consider the following toy model of the Internet, containing 6 interlinked pages as shown in Figure 1. The transition probabilities are then given by the matrix $A = (1 - \alpha)B + \alpha C$ where:

$$B = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$\alpha = 0.15$ and $C = (1/6)^{6\times 6}$.

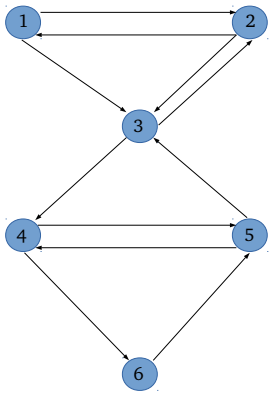

Figure 1: Toy model of the Internet with 6 pages.

i) Compute the stationary π distribution of the above Markov chain.

ii) Assume an action to represent displaying an ordered set of 3 pages to a user. The reward of an action is 3 if the first page in the list is relevant, 2 if the second item is the first relevant item, 1 if the last item is the only relevant one and 0 if none of the items are. Considering a model where a page k is relevant with probability π_k, which is the action with the highest expected reward?

iii) You are the administrator of web page 3 in Figure 1. Due to space constraints, on the mobile version of your web page, you must remove one of your outgoing hyperlinks. Knowing that

after leaving your page users follow at most 4 links before quitting, which link do you keep in order to maximize the probability that users leaving your web page return? (assume $\alpha = 0$)

2.4

Assume a small computer network interconnected according the graph shown in Figure 2. In this figure, the weight of the edge between two nodes k_1 and k_2, which we denote by $\theta_{k_1 k_2}$, represent the success probability of a transmission from one node to another. We consider the problem of transmitting a packet from node 1 to node 6. Time proceeds in rounds, at each round t, the node $n(t)$ holding the packet will attempt a transmission to one of its neighbors. The success of a transmission at time t from a node k_1 to a node k_2 is given by a Bernoulli random variable $X(t)$ with $\mathbb{E}[X(t)] = \theta_{k_1 k_2}$. If the transmission from k_1 to k_2 is successful (i.e. $X(t) = 1$), in the next round $t + 1$, we will have $n(t + 1) = k_2$, otherwise, the packet is lost forever (no retransmission), $n(t+1) = 0$. Since we start from node 1, $n(0) = 1$. You get to decide the outgoing edges of each of the nodes in the network i.e. for every node k_1 you must decide to which neighbor it will forward received packets. You receive a reward of 1 if the packet reaches node 6 in at most 4 rounds and 0 otherwise. Your goal is to maximize the probability of a packet reaching node 6.

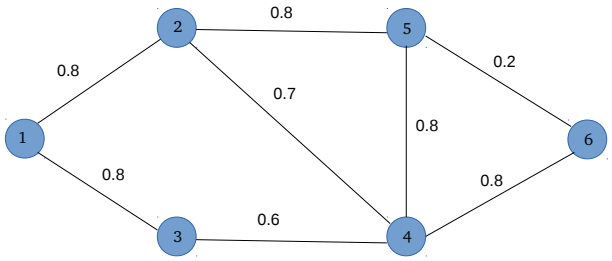

Figure 2: Network containing 6 nodes.

i) Formulate the above problem as a finite time MDP. What is the time horizon?

ii) Compute the optimal policy π with the maximal expected reward.

2.5 (Dynamic Programming)

You are given 2 identical eggs and a 100 floor high tower. For some reason, you are tasked with determining the hardness of the egg shells by finding the highest floor you can drop the eggs from, without them breaking while using the fewest number of drops **in the worst case**. The highest floor from which the eggs can be dropped without breaking now becomes our standard measure of egg hardness. The hardness of the shells in your batch is unknown 1 and 100. Note that if an egg did not break after a drop, it can be re-used.

2.6

A thief goes out and robs a house every night. If he is caught, which happens with probability p, he loses all his money. If he is not caught, the value of the things he manages to steal is added to his fortune. Assume that he focuses his nightly raids on one part of town, so that the sum of valuables in each house is i.i.d. He can at any night choose to retire, and then keep his previous earnings. To help the thief plan his retirement, model his situation as an MDP.

2.7

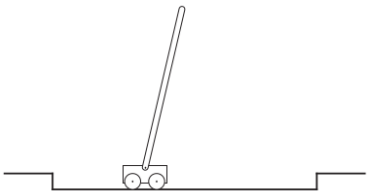

Figure 3: Pole-cart system. Figure from [3].

Consider the pole-cart setup in Figure 3. You can apply no force, or a unit force in either the left or right direction to the cart. The goal is to find a policy that balances the pole in an upright position. Model this as an MDP. (Treat the problem as a learning problem; you do *not* need to specify the transition probabilities.) The distance between the left and right wall is L.

2.8

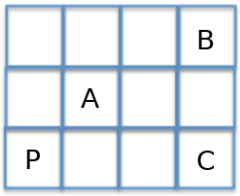

Figure 4: Simple gridworld example.

Consider a person (P) moving on the rectangular grid shown in Figure 4. At each time-step, the person can decide to move either up, down, left or right. The map "wraps around" at the borders, meaning that, for example, if the player moves up when in a square in the top row, he is teleported to the square in the same column in the bottom row. If he enters the square marked A, he receives a reward +1 and is teleported with probability 1/3 and 2/3 to one of the squares B and C, respectively, on the next move. His goal is to find a policy that optimizes a discounted long-term reward.

2.9

An office has bought a mobile robot whose task is to collect empty (and dirty) coffee cups. It has sensors to detect such cups, as well as a gripper-arm that it can use to pick these up and place on an onboard tray. The robot runs on a rechargeable battery with two possible charge-levels: low and high. For each period (= 10 min) of its high-level planning algorithm, the robot can either *i)* actively search for an empty cup, *ii)* remain stationary and wait for someone to place a cup on it, *iii)* head back to its recharging station to recharge its battery.

The best way to collect cups is to search for them, but this runs down the robot's battery level, whereas waiting does not. More precisely, when the robot decides to search for cups, if the charge level is *high*, then a period of active search can always be completed without risk of depleting the battery. However, at the end of the period the battery is reduced to the *low* charge-level with probability $\alpha > 0$. On the other hand, a period of search initiated with a *low* charge-level leaves the energy level at *low* with probability $\beta > 0$, and completely depletes the battery with probability $1 - \beta$. In the latter case, the robot must be "rescued" and the battery is recharged back to *high*.

Each cup collected counts as a unit reward, whereas being rescued inquires a negative reward -3. Denote by r_{search} and r_{wait} (with $r_{\text{search}} > r_{\text{wait}}$) the expected number of cups the robot will collect while searching and waiting, respectively. Finally, assume that no cups can be collected when the robot is going back to its recharging station, or on a step when the battery is completely depleted

and the robot is being rescued. Since the office is very crowded, if the robot needs rescue, it will get rescued within one period of time. The robot aims to maximize its performance on each working day (= 8 hours) independently.

Provide the MDP that the robot's manufacturer has used (based on the description above) to compute an optimal policy.

Note: You might have to model the rewards as depending also on the next state: $r_t(s_t, a_t, s_{t+1})$.

2.10 (Based on [5])

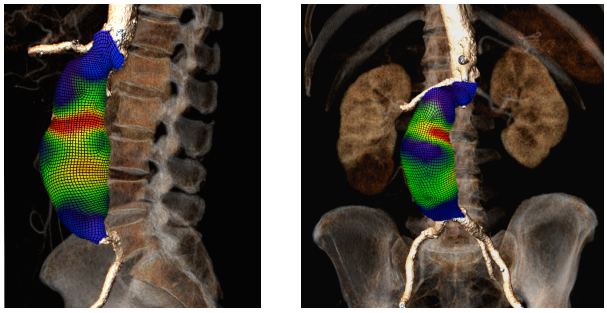

Figure 5: A 3D reconstruction of an AAA seen from a lateral (left) and a forward-facing (right) perspective. The AAA is clearly marked by the difference in coloring.

An *abdominal aortic aneurysm* (AAA) is an enlargement of the abdominal aorta (see Figure 5) that is asymptomatic, but that can rupture with fatal consequences. The risk of rupture as well as its growth rate are highly related to the diameter (size) of the AAA. It is possible to treat an AAA: a doctor can perform surgery with the aim of replacing the diseased part of the aorta with a synthetic stent. However, this operation is not without risk. In particular, the older a patient is, the higher the risk of complications in the surgery. The doctor would like to know, as a function of the patient's age and AAA size, what the optimal action is (to perform surgery, or not) so as to maximize the life expectancy of the patient. Assume that a decision has to be made every year. The typical size of an AAA is between 3 cm and 10 cm.

a) Help the doctor by modeling this as an MDP.

b) What does your intuition say regarding the structure of the optimal policy?

c) Can you think of any extensions that would make your model more realistic?

2.11

Devise three examples/applications/tasks that fit the MDP framework on your own, and provide the corresponding MDP models. Try to make the three examples as *different* from each other as possible.

2.12

This exercise investigates the relationship between control theory and reinforcement learning.

Consider a linear stochastic control system over $\mathbb{Z}^n$. Such a system is defined by a state $s_t \in \mathbb{Z}^n$ with stochastic dynamics given by the following recursion:

$$s_{t+1} = As_t + Ba_t + w_t, \qquad s_0 = 0,$$

with matrices $A \in \mathbb{Z}^{n \times n}$ and $B \in \mathbb{Z}^{n \times m}$.

Furthermore, we shall assume that the w_t are iid and uniformly distributed over $(\pm e_i)_{i=1}^n$, the standard basis vectors of $\mathbb{Z}^n$. That is,

$$\mathbb{P}\left(w_t = \begin{bmatrix} 0 & \dots & 0 & \underbrace{\pm 1}_{\text{in the } i\text{th position}} & 0 & \dots & 0 \end{bmatrix}^\top\right) = \frac{1}{2n}, \text{ with } i = 1, \dots n.$$

Assume that the actions are restricted to $\mathbb{Z}^m$ and that they are only allowed to depend on past states. Finally, the control cost is given by

$$c_t(s_t, a_t) = \mathbb{E}[s_t^\top s_t] + \mathbb{E}[a_t^\top a_t].$$

a) What is the state space? For $n = 2$ draw a picture.

b) Show that for $a_t = 0$ for all t, this is a markov chain

c) In the general case with a_t allowed to depend on the entire history, formulate this as an MDP and describe the transition probabilities.

3 Solutions

Solution to Problem 2.1

Part (a)

The state at time t can be described by t and the number of times head n has been observed. In addition we set state $\emptyset$ to indicate that the decision maker stopped the process. Hence for example, $S = (\{1, \ldots, T\} \times \{1, \ldots, T\}) \cup \emptyset$ (the first coordinate represents time and the second n). There are two actions, continue (C) or stop (S). The reward function is as follows: for all $1 \leq t \leq T$, $r((t,n), S) = n/t$, $r((t,n), C) = 0$; when in state $\emptyset$, the reward is 0, i.e., $r(\emptyset, x) = 0$ for $x = C$ or S. The transition probabilities are $p(\emptyset|(t,n), S) = 1$, $p(\emptyset|\emptyset, x) = 1$ for $x = C$ or S, and $p((t+1, n+1)|(t,n), C) = 1/2 = p((t+1, n)|(t,n), C)$. We have a finite horizon MDP (without discount).

Part (b)

The state at time t (where time is interpreted as the number of times the dictor has tried to pass a law) can be described by t and the number of laws n that have passed parliament. In addition we set state $\emptyset$ to indicate that the dictator stopped trying to pass laws and X denoting the revolt. Hence $S = (\{0, \ldots, N\} \times \{0, \ldots, N\}) \cup \emptyset \cup \{X\}$ (the first coordinate represents time and the second n). There are two actions, continue ($\mathcal{C}$) or stop ($\mathcal{S}$). The reward function is as follows: for all $0 \leq t \leq N$, $r((t,n), \mathcal{S}) = w_0 2^n$, $r((t,n), \mathcal{C}) = 0$; when in state $\emptyset$, the reward is 0, i.e., $r(\emptyset, x) = 0$ for $x = \mathcal{C}$ or $\mathcal{S}$. The transition probabilities are $p(\emptyset|(t,n), \mathcal{S}) = 1$, $p(\emptyset|\emptyset, x) = 1$ for $x = \mathcal{C}$ or $\mathcal{S}$, and $p((t+1, n+1)|(t,n), \mathcal{C}) = (1-p_p)(1-p_r)$, $p((t+1, n)|(t,n), \mathcal{C}) = p_p(1-p_r)$, $p(X|(t,n), \mathcal{C}) = p_r$, $p(X|X, x) = 1$, $x = \mathcal{C}$ or $\mathcal{S}$. We have a finite horizon MDP (without discount).

Part (c)

The state at time t can be described as the sum of observations up to time t, which we denote by s. In addition, we use $\emptyset$ to indicate that the decision maker stopped the process. The state space is then $S = \{1, ..., 6T\} \cup \emptyset$. The actions are (C) (continue) and (S) (stop). The reward function is given by $r(s, C) = 0$ and $r(s, S) = s - 21$ if $s \leq 21$ and $r(s, S) = -10$ otherwise. The reward in state $\emptyset$ is always 0, $r(\emptyset, x) = 0$ for $x = C$ or S. The transition probabilities are given by $p(\emptyset|s, (S)) = 1$, $p(\emptyset|\emptyset, x) = 1$ for $x = C$ or S and, finally, $p(s+i|s, C) = 1/6$, for $i = 1, ..., 6$. We have a finite horizon MDP (without discount).

Solution to Problem 2.2

Let the maximum bid be $10\,000 \cdot k_{\max} \in \mathbb{Z}$. Then, one model is:

- **State-space:** $S = \{10\,000 \cdot k \text{ for } 0 \leq k \leq k_{\max} \text{ and } k \in \mathbb{Z}\} \cup \{\text{Sold}\}$, where a state $s \neq \text{Sold}$ represents the current bid under consideration.
- **Actions:** $\mathcal{R}$ – Reject, $\mathcal{S}$ – Sell (accept)
- **Time-horizon and objective:** Finite horizon N, $\mathbb{E}\{\sum_{t=0}^{N-1} r_t(s_t, a_t) + r_N(s_N)\}$
- **Rewards:** Terminal: $r_N(\text{Sold}) = 0$, $r_N(x) = x$ for $x \in \{10\,000 \cdot k \text{ for } 0 \leq k \leq k_{\max} \text{ and } k \in \mathbb{Z}\}$. Non-terminal: $r_t(s = x, a = \mathcal{S}) = x \cdot (1+\rho)^{N-t}$, $r_t(s = \text{Sold}, a = \mathcal{S}) = r_t(s = \text{Sold}, a = \mathcal{R}) = r_t(s = x, a = \mathcal{R}) = 0$.
- **Transition probabilities:** The non-zero transitions are
  - $p_t(s' = 10\,000 \cdot k \big| s \in S \setminus \{\text{Sold}\}, a = \mathcal{R}) = \Pr\{w_t = 10\,000 \cdot k\}$, for $k \in \mathbb{Z}$
  - $p_t(s' = \text{Sold} \big| s \in S \setminus \{\text{Sold}\}, a = \mathcal{S}) = 1$
  - $p_t(s' = \text{Sold} \big| s = \text{Sold}, a = \cdot) = 1$

Solution to Problem 2.3

Part i)

Simple computation gives the transition probabilities of our Markov chain as:

$$A = \begin{pmatrix} 0.025 & 0.45 & 0.45 & 0.025 & 0.025 & 0.025 \\ 0.45 & 0.025 & 0.45 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.45 & 0.025 & 0.45 & 0.025 & 0.025 \\ 0.025 & 0.025 & 0.025 & 0.025 & 0.45 & 0.45 \\ 0.025 & 0.025 & 0.45 & 0.45 & 0.025 & 0.025 \\ 0.025 & 0.025 & 0.025 & 0.025 & 0.875 & 0.025 \end{pmatrix},$$

We can then extract the stationary distribution π of the Markov chain by solving the following equation $\pi A = \pi$. This yields approximately $\pi = (0.092, 0.158, 0.220, 0.208, 0.209, 0.113)$.

Part ii)

Having identified the stationary distribution, the action with the maximal expected reward is simply the list u containing the pages k maximizing π_k, in descending order of π_k:

$$u = (3, 5, 4).$$

Part iii)

Denote by d the number of hops until a user returns. We proceed to compute the probability that $d < 5$, starting with the hyperlink towards page 2:

$$\begin{aligned} \mathbb{P}_2[d = 2] &= 1/2 (\text{probability of returning from page 2 directly to 3}) \\ \mathbb{P}_2[d = 3] &= 1/2 \times 1/2 \text{ (probability of going from page 2 to 1 then 3)} \\ \mathbb{P}_2[d = 4] &= 1/2 \times 1/2 \times 1/2 \text{ (probability of going from page 2 to page 1} \\ &\qquad \text{and then return to page 2 then page 3)} \end{aligned}$$

hence we have $\mathbb{P}_2[d \leq 4] = 7/8$. Now for the action of keeping the hyperlink towards page 4:

$$\begin{aligned} \mathbb{P}_4[d = 2] &= 0 \\ \mathbb{P}_4[d = 3] &= 1/2 \times 1/2 \\ \mathbb{P}_4[d = 4] &= 1/2 \times 1/2, \end{aligned}$$

hence we have $\mathbb{P}_2[d \leq 4] = 1/2$. Consequently, we should keep the link towards page 2.

Solution to Problem 2.4

Part i)

The states are represented by the current node storing the packet. The actions at a state k are given by the immediate neighbors of k in the graph in Figure 2 (we must choose through which neighbor we should route the packet). We denote by $a_{k_1 k_2}$ the action of attempting to transmit the packet from k_1 to k_2. The transition probability from a state k_1 to a state y under action $a_{k_1 k_2}$ is then given by:

$$p(y|k_1, a_{k_1 k_2}) = \begin{cases} \theta_{k_1 k_2}, \text{ if } y = k_2, \\ 1 - \theta_{k_1 k_2}, \text{ if } y = 0, \\ 0, \text{ otherwise.} \end{cases}$$

Part ii)

We use Dynamic Programming (DP). In the first step of the algorithm we compute the best path (i.e. the one maximizing the probability of successful transmission) between nodes immediately

adjacent to the destination node 6 and the destination node. We denote by $d(x, y)$ the transmission success probability of the best path between x and y. We then have:

$$d(5, 6) = 0.64 \text{ and } d(4, 6) = 0.8.$$

For the next step of the algorithm, we look at the immediate neighbors of nodes 4 and 5:

$$d(2, 6) = \max(0.8 \times d(5, 6), 0.7 \times d(4, 6)) = 0.56 \text{ and } d(3, 6) = 0.6 \times d(4, 6) = 0.48.$$

Finally, we look at the source node:

$$d(1, 6) = \max(0.8 \times d(2, 6), 0.8 \times d(3, 6)) = \max(0.448, 0.384) = 0.448.$$

Consequently, the optimal path through the network is $1 \to 2 \to 4 \to 6$.

Solution to Problem 2.5

We encode the state of the DP model by pairs (n, k) $(k, n \geq 1)$, where n represents the number of floors still needing to be explored and k the number of eggs still available. Denoting by $f(n, k)$ the number of drops required solve the problem starting from state (n, k) **in the worst case** we then have:

$$f(n, k) = 1 + \min_{x=1,..,n} \max(f(x - 1, k - 1), f(n - x, k)).$$

The above equation stems from the states being achieved by dropping the egg from floor x and it breaking (i.e. $(x - 1, k - 1)$) and not breaking (i.e. $(n - x, k)$). We take the maximum of the two costs as we can control x but nature is adversarial so nature will control whether the egg breaks or not such that it maximizes the number of drops required.

Now we determine the minimized of the above cost function. Imagine the first drop occurs at a floor n. If the egg breaks, we would need to proceed with the second egg floor by floor, starting from floor 1. This will result in a total of at most n total drops. If it does not break, we have 1 play to account for and restart the problem with the $100 - n$ remaining floors above n, but now dropping the first egg from $n - 1$ floors above n. Again, if the egg breaks we must continue floor-by-floor with the second egg, resulting in $n - 1$ drops plus an additional 1 from the previous attempt, so again, a total of n drops. Hence we must minimize n under the constraint that we cover all floors. In order to cover all floors we must pick n such that:

$$n + (n - 1) + (n - 2) + ... + 1 = n(n + 1)/2 \geq 100,$$

Hence, our desired minimizer is $n = 14$. It remains to prove the optimality of our policy.

Proof. Denote by x^* the lowest floor from which the egg breaks when dropped. Define $f_\pi(n, k)$ the number of drops required to solve the problem in the worst case under policy π. Assume the true optimal policy π executes the first drop from a level n. We proceed to show that if the egg survives and the second drop is not executed from the level $2n-1$, we will have $f^\pi(100, 2) \geq f(100, 2)$, where $f(100, 2)$ being the cost achieved by the policy obtained according to the procedure described in our solution.

Note that the interval between two drops under the optimal solution is **at most** 14, otherwise, in the worst case we will have $f^\pi(100, 2) > f(100, 2) = 14$. Let i index the number of drops and denote by $x^\pi(i)$ the number of floors between the floor from which we drop the egg at round i and the highest floor from which the egg survived up to round i, under policy π. Now, it there exists $i > 1$ such that $x^\pi(i) \geq 14$ we will then have that in the worst case $f^\pi(100, 2) \geq 2+13 > f(100, 2) = 14$. Similarly, for any i, if $x^\pi(i) > 14 - (i - 1)$ we will have $f^\pi(100, 2) > i + x^\pi(i) - 1 = 14$ in the worst case. We have so far established that for all i, the optimal policy π satisfies $x^\pi(i) \leq 14 - (i - 1)$.

Furthermore, in order for $f^\pi(100, 2) < f(100, 2) = 14$ we must have that $\forall i$, $x^\pi(i) < 14 - (i - 1)$, as we are interested in the **worst case**. Consequently, the starting floor is $n \leq 13$. Since $x^\pi(i) \in \mathbb{N}$ we must have that $\forall i$, $x^\pi(i) \leq 13 - (i - 1)$ and hence the number of floors N^π covered by such a policy until $x^\pi(i) < 1$ (i.e. in the first n drops), if $x^* = 100$, is:

$$n + n - 1 + ... + 1 = n(n + 1)/2 \leq 13(13 + 1)/2 = 91.$$

Hence, since after n drops the policy π can only proceed floor by floor, we must have that $f^\pi(100,2) \geq n + 100 - n(n+1)/2 \geq 13 + 100 - 13(13+1)/2 = 13 + 9 = 22 > f(100,2) = 14$, with the first inequality stemming from $n + 100 - n(n+1)/2$ being monotonically decreasing in n, for all $n \leq 100$, $n \in \mathbb{N}$, which concludes the proof.

Solution to Problem 2.6

One interpretation of the infinite-horizon discounted objective function is that the decision maker optimizes over a (random) geometrically distributed time horizon T. That is, the system "stays alive" with probability equal to the discount factor, if it is currently "alive".

In this problem, we interpret "staying alive" as not getting caught and avoiding prison. Let w_t be the value of the valuables in the house robbed on night t. The MDP is then:

- **State-space:** $S = \mathbb{R}_{\geq 0} \cup \{\text{Retired}\}$, where a state $s \in \mathbb{R}_{\geq 0}$ represents the wealth he has collected this far.
- **Actions:** $\mathcal{R}$ – Retire, $\mathcal{C}$ – Continue
- **Rewards:** $r_t(s = x, a = \mathcal{R}) = x$ for $x \in \mathbb{R}_{\geq 0}$, all other zero: $r_t(s = x, a = \mathcal{C}) = r_t(s = \text{Retired}, a = \cdot) = 0$.
- **Time-horizon and objective:** Discounted infinite horizon, $\mathbb{E}\big\{\sum_{t=0}^{\infty}(1-p)^t r_t(s_t, a_t)\big\}$
- **Transition probabilities:** The only non-zero transitions are
  - $p_t(s' = \text{Retired} | s = \text{Retired}, a = \cdot) = 1$
  - $p_t(s' = \text{Retired} | s = x, a = \mathcal{R}) = 1$
  - $p_t(s' = y | s = x, a = \mathcal{C}) = \Pr\{w_t = y - x\}$

Solution to Problem 2.7

Let x be the distance of the cart from the left wall, and θ the angle of the pole (measured, say, counter-clockwise from the positive horizontal axis). Denote the velocity of the cart as $\dot{x}$ and the angular velocity of the pole $\dot{\theta}$.

- **State-space:** $s_t = (x, \dot{x}, \theta, \dot{\theta}) \in S = [0, L] \times \mathbb{R} \times [0, 2\pi] \times \mathbb{R}$.
- **Actions:** $a_t \in \mathcal{A} = \{-1, 0, 1\}$
- **Rewards:** Give a reward -1 whenever the pole is not in a sufficiently upright position: $r(s_t, \cdot) = -1$ if $\theta \notin [\frac{\pi}{4}, \frac{3\pi}{4}]$, zero otherwise.
- **Time-horizon and objective:** Discounted infinite horizon, $\mathbb{E}\big\{\sum_{t=0}^{\infty}\lambda^t r_t(s_t, a_t)\big\}$
- **Transition probabilities:** These encode the dynamics of the system. To derive these, take the course *EL2820 Modelling of Dynamical Systems*. Later in this course, we will try to learn these directly from the environment (in a computer lab).

Solution to Problem 2.8

Let x and y denote the player's column and row, respectively, measured from the bottom-left corner.

- **State-space:** $s_t = (x, y) \in S = \{1, 2, 3, 4\} \times \{1, 2, 3\}$.
- **Actions:** $a_t \in \mathcal{A} = \{\text{up, down, left, right}\}$
- **Rewards:** $r(s_t, \cdot) = +1$ if $x = 2$ and $y = 2$ (shorthand: if $s_t = A$), zero otherwise.
- **Time-horizon and objective:** Discounted infinite horizon, $\mathbb{E}\big\{\sum_{t=0}^{\infty}\lambda^t r_t(s_t, a_t)\big\}$

- **Transition probabilities:** We state the non-zero transitions. First, if the player stands in the teleportation square:
  - $p_t(s' = B|s = A, a = \cdot) = 1/3$
  - $p_t(s' = C|s = A, a = \cdot) = 2/3$

  Otherwise, (i.e., if $(x, y) \neq (2, 2)$):
  - $p_t(s' = (x, y + 1)|s = (x, y), a = \text{up}) = 1$ if $y \neq 3$.
  - $p_t(s' = (x, 1)|s = (x, 3), a = \text{up}) = 1$.
  - $p_t(s' = (x, y - 1)|s = (x, y), a = \text{down}) = 1$ if $y \neq 1$.
  - $p_t(s' = (x, 3)|s = (x, 1), a = \text{down}) = 1$.
  - $p_t(s' = (x + 1, y)|s = (x, y), a = \text{right}) = 1$ if $x \neq 4$.
  - $p_t(s' = (1, y)|s = (x, 4), a = \text{right}) = 1$.
  - $p_t(s' = (x - 1, y)|s = (x, y), a = \text{left}) = 1$ if $x \neq 1$.
  - $p_t(s' = (4, y)|s = (1, y), a = \text{left}) = 1$.

Problems that have this structure are commonly referred to as *gridworld* problems.

Solution to Problem 2.9

- **State-space:** $S = \{\text{low}, \text{high}\}$.
- **Actions:** In general, $\mathcal{A} = \{\text{search, wait, recharge}\}$. However, we can incorporate some prior knowledge (that it's foolish to recharge if our battery is full): $\mathcal{A}(\text{high}) = \{\text{search, wait}\}$ and $\mathcal{A}(\text{low}) = \{\text{search, wait, recharge}\}$.
- **Transition probabilities:** The non-zero transitions are:
  - $p_t(s' = \text{high}|s = \text{high}, a = \text{wait}) = 1$
  - $p_t(s' = \text{high}|s = \text{high}, a = \text{search}) = 1 - \alpha$
  - $p_t(s' = \text{low}|s = \text{high}, a = \text{search}) = \alpha$
  - $p_t(s' = \text{low}|s = \text{low}, a = \text{wait}) = 1$
  - $p_t(s' = \text{low}|s = \text{low}, a = \text{search}) = \beta$
  - $p_t(s' = \text{high}|s = \text{low}, a = \text{search}) = 1 - \beta$ (Robot was rescued)
  - $p_t(s' = \text{high}|s = \text{low}, a = \text{recharge}) = 1$
- **Rewards:** We specify only the rewards for transitions that have non-zero probability:
  - $r_t(s = \text{high}, a = \text{search}, s' = \text{high/low}) = r_{\text{search}}$
  - $r_t(s = \text{high}, a = \text{wait}, s' = \text{high}) = r_{\text{wait}}$
  - $r_t(s = \text{low}, a = \text{wait}, s' = \text{low}) = r_{\text{wait}}$
  - $r_t(s = \text{low}, a = \text{search}, s' = \text{low}) = r_{\text{search}}$
  - $r_t(s = \text{low}, a = \text{search}, s' = \text{high}) = -3$ (Robot was rescued)
  - $r_t(s = \text{low}, a = \text{recharge}, s' = \text{high}) = 0$ (Cannot collect anything when recharging)

- **Time-horizon and objective:** Finite-horizon with $T = 8{\times}6 = 48$: $\mathbb{E}\{\sum_{t=0}^{T} r_t(s_t, a_t, s_{t+1})\}$

Solution to Problem 2.10

Part a)

- **State-space:** We simplify by discretizing the aneurysm size (which really is a continuous variable), and introduce a terminal state as well as a state after successful surgery: $S = \{3-4 \text{ cm}, 4-5 \text{ cm}, \ldots, 9-10 \text{ cm}, \text{dead}, \text{healthy}\}$.
- **Actions:** $\mathcal{O}$ – Operate (i.e., surgery), $\mathcal{N}$ – Nothing
- **Rewards:** In order to maximize life-expectancy, give reward 1 for every year the patient is alive: $r_t(s = \text{dead}, a = \cdot) = 0$, all other 1.
- **Time-horizon and objective:** We identify time t as the age of the patient. Hence, we have a finite horizon problem with, say, $T = 120$: $\mathbb{E}\{\sum_{t=0}^{120} r_t(s_t, a_t)\}$
- **Transition probabilities:** A very simplistic model of the action of events is as follows. Let us first introduce a number of parameters: Denote the probability of death by natural causes at age t as d_t, the probability of failure in surgery as $f_t(\text{size})$, the probability of a rupture as $R(\text{size})$ and the probability of growing one size $g(\text{size})$. The non-zero transitions are then:
  - $p_t(s' = \text{size} | s = \text{size}, a = \mathcal{N}) = (1 - d_t) \times (1 - g(\text{size})) \times (1 - R(\text{size}))$
  - $p_t(s' = \text{next size} | s = \text{size}, a = \mathcal{N}) = (1 - d_t) \times g(\text{size}) \times (1 - R(\text{size}))$
  - $p_t(s' = \text{dead} | s = \text{size}, a = \mathcal{N}) = d_t + R(\text{size}) - d_t \times R(\text{size})$
  - $p_t(s' = \text{healthy} | s = \text{size}, a = \mathcal{O}) = 1 - f_t(\text{size})$
  - $p_t(s' = \text{dead} | s = \text{size}, a = \mathcal{O}) = f_t(\text{size})$
  - $p_t(s' = \text{healthy} | s = \text{healthy}, a = \cdot) = 1 - d_t$
  - $p_t(s' = \text{dead} | s = \text{healthy}, a = \cdot) = d_t$
  - $p_t(s' = \text{dead} | s = \text{dead}, a = \cdot) = 1$

  Note that the transition probabilities are time-*variant* (since we have dependency on patient age).

Part b)

Since i) the surgery gets more dangerous, and ii) the AAA grows in size as the patient ages, surgery is to be preferred early when an AAA is detected. There is, however, a trade-off between remaining life-expectancy and the surgical risk (that is parameter dependent) that also has to be taken into account. A (time-dependent) threshold policy can be expected.

Part c)

Some examples:

- Include discount-factor (you, commonly, get sicker with older age)
- Different treatment procedures
- The action set can be state/age dependant
- ...

Solution to Problem 2.12

a)

The state space is $\mathbb{Z}^n$, for $n = 2$ this can be visualized as an infinite square grid.

b)

Observe that $s_{t+1} = F(s_t, w_{t+1}) = As_t + w_{t+1}$ and use the result from Exercise 1.13 of the last sheet.

c)

Observe that as in a) the state space is $S = \mathbb{Z}^n$, the possible values of s_t. As for a_t, B is an $m \times n$ matrix, so its inputs are vectors of dimension m, wherefore we conclude that the action space is $A = \mathbb{Z}^m$.

To find the transition kernel p, denote by $q(\cdot)$ the probability mass function of w_t. Now,

$$\begin{aligned} p(s'|a,s) & \\ &= \mathbb{P}(s_{t+1} = s'|a_t = a, s_t = s) \\ &= \mathbb{P}(As_t + Ba_t + w_{t+1} = s|a_t = a, s_t = s) \\ &= \mathbb{P}(w_{t+1} = s - As_t - Ba_t|a_t = a, s_t = s) \\ &= \mathbb{P}(w_{t+1} = s - As - Ba) \\ &= q(s' - As - Ba) \end{aligned}$$

Finally, the reward is just the negative of the control cost,

$$r_t(s_t, a_t) = -c_t(s_t, a_t) = -\mathbb{E}[s_t^\top s_t] - \mathbb{E}[a_t^\top a_t].$$

Observe that actually $r_t = r$, i.e. the reward function is homogenous.

References

[1] D. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific, 1995.

[2] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[4] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

[5] R. Mattila, A. Siika, J. Roy, and B. Wahlberg, "A Markov decision process model to guide treatment of abdominal aortic aneurysms," in *Proceedings of the IEEE Conference on Control Applications (CCA'16)*, pp. 436–441, 2016.