

## CASE STUDY: AIRLINE TICKET SALES

### 1. PROBLEM

Assume flight AA 001 departs in the evening of May 15. It has seat capacity 10. The airline starts to sell tickets on May 1 and ends on May 15. So, there are 15 days on which tickets to available seats can be sold. On each day two types of tickets can be sold: *cheap* for \$100 and *expensive* for \$200. On each day airline allocates at most two tickets for sale, and it has to decide how many cheap and expensive tickets (out of the total of at most 2) should be offered for sale on each of the 15 days. On any day, the airline cannot allocate more tickets for sale than there are unsold tickets.

It is known that the number of ticket requests that arrive on each day is random and has the following probability distribution:

$$\text{number of ticket requests is } \begin{cases} 2 & \text{with probability } 1/4, \\ 1 & \text{with probability } 1/2, \\ 0 & \text{with probability } 1/4. \end{cases}$$

It is also known that the numbers of requests are independent across days.

On each day, here is how the ticket sale works. If the number of requests exceeds the number of offered tickets, the excess requests are discarded. The remaining requests (whose number does not exceed the number of allocated tickets for sale), first, certainly buy whatever cheap tickets available and, second, leftover requests buy expensive tickets with probability 0.4 and don't buy with probability 0.6.

Example 1: Airline allocates one cheap and one expensive ticket for sale on a given day. If two requests arrive, one certainly buys the cheap ticket and another one buys expensive ticket with prob. (w.p.) 0.4 and does not buy anything w.p. 0.6. If one request arrives, it will buy cheap ticket. If no requests arrive, nobody buys tickets, of course.

Example 2: Airline allocates one expensive ticket for sale on a given day. If two requests arrive, one certainly goes away without buying anything, and another one buys the expensive ticket with prob. (w.p.) 0.4 and does not buy anything w.p. 0.6. If one request arrives, it buys the expensive ticket with prob. (w.p.) 0.4 and does not buy anything w.p. 0.6.

Example 3: Airline allocates two expensive tickets for sale on a given day. If two requests arrive, each of them will buy an expensive ticket with prob. (w.p.) 0.4 and will not buy anything w.p. 0.6. If one request arrives, it buys an expensive ticket with prob. (w.p.) 0.4 and does not buy anything w.p. 0.6.

Example 4: Airline allocates one cheap ticket for sale on a given day. If two

requests arrive, one certainly buys the cheap ticket and another goes away without buying anything. If one request arrives, it buys the cheap ticket.

**Project goal:** Find an optimal ticket sale policy, to maximize the expected total revenue. No discount, i.e. the discount factor  $\alpha = 1$ .

## 2. MODEL AND YOUR ASSIGNMENT

Ticket sale process is an MDP  $X_n$ , where  $n$  is time. State is the number of remaining (unsold) tickets. State space:  $\{0, 1, \dots, 10\}$ . Time horizon:  $n = 0, 1, \dots, 14$ , where  $n = 0$  corresponds to May 1 and  $n = 14$  corresponds to May 15. Action is  $k = (p, q)$ , where  $p$  and  $q$  is the number of cheap and expensive tickets offered, respectively. Note that  $p, q \geq 0$ ,  $p + q \leq 2$ . Possible actions:  $A = \{(0, 0), (1, 0), (0, 1), (2, 0), (1, 1), (0, 2)\}$ . Note that not all actions can be taken in each state: we cannot allocate more tickets for sale than the number of tickets left; *you have to take this into account*. Remark: For coding purposes it is convenient to enumerate actions  $(0, 0), (1, 0), (0, 1), (2, 0), (1, 1), (0, 2)$  by, for example,  $0, 1, 2, 3, 4, 5$ . However, for the purposes of explanation, I will keep referring to an action  $k$  as a pair  $(p, q)$ .

What is the reward corresponding to each action  $k$ ? In this case, it will be the *expected revenue* produced by this action. Denote:  $g_k(i)$  = the reward (expected revenue) if action  $k$  is taken in state  $i$ . Again, not every action can be taken in every state. If state  $i \geq 2$ , any action can be taken; if state  $i = 1$ , only actions  $(0, 0), (1, 0), (0, 1)$  are available; if state  $i = 0$ , only action  $(0, 0)$  is available. Clearly, *in this particular model*,  $g_k(i)$  does not depend on  $i$  as long as action  $k$  is available in state  $i$ .

For example, what is the reward  $g_{(1,1)}$  of action  $(1, 1)$ ? (Once again, this action can only be taken in state  $i \geq 2$ .)

$$g_{(1,1)} = (1/4) * 0 + (1/2) * 100 + (1/4) * [100 + 0.4 * 200 + 0.6 * 0] = 95.$$

**You have to compute the rewards for all other actions.**

What about transition probabilities  $P_k(i, j)$  for each action  $k$ ? Yet again, action  $k = (p, q)$  can be applied in state  $s$  only if  $i \geq p + q$ . If the number of tickets for sale  $p + q = 2$ , then transitions from  $i$  are possible to states  $i, i - 1, i - 2$ . For example, if  $k = (1, 1)$

$$P_{(1,1)}(i, i) = 1/4 = 0.25; \quad P_{(1,1)}(i, i - 1) = 1/2 + (1/4) * 0.6 = 0.65;$$

$$P_{(1,1)}(i, i - 2) = (1/4) * 0.4 = 0.1.$$

**You have to compute the transition probabilities for all actions.**

If the number of tickets for sale  $p + q = 1$ , then transitions from  $i$  are possible to states  $i, i - 1$ .

**You have to compute the transition probabilities for all actions.**

If the number of tickets for sale  $p + q = 0$ , i.e. the action is  $(0, 0)$ , then, of course  $P_{(0,0)}(i, i) = 1$ .

**Main assignment:** Write a python code, which computes the optimal policy, maximizing the total expected revenue. (You also have to calculate

the rewards and transition probabilities above – but for that you do not have to write a code, can just compute and explain numbers in your report.) The output of the code has to show an optimal policy, i.e. action which should be taken, depending on time and state; a convenient way to show this is a matrix. It also has to show the optimal value function; this is also convenient to show as a matrix.

You must submit BOTH the report explaining what you do and how, and the working code. Just a report, without code, does NOT give you a partial credit.