

## Pandora's Box

Thomas Kesselheim

Last Update: May 18, 2020

Today, we will get to know another example of a stochastic problem, which can be captured by a Markov decision process with a huge state space. The optimal policy, however, is not difficult to compute. It is, in fact, a good representative for many similar problems.

## 1 Problem Statement

We have  $n$  boxes. Each of the boxes contains a prize of a certain value. We may open as many boxes as we like. However, opening a box costs a certain amount. We are only allowed to take home a single one of the prizes. We may adapt our choices depending on what we find in the boxes that we open.

More formally, box  $i$  contains a prize of value  $v_i$ . We don't know  $v_i$  but only its distribution until we open the box. Opening box  $i$  costs  $c_i$ . The final reward is given as

$$\max_{i:\text{box } i \text{ opened}} v_i - \sum_{i:\text{box } i \text{ opened}} c_i ,$$

where we define the maximum as 0 if no boxes are opened.

We skip the detailed description as a Markov decision process this time. You should notice that we need to store the maximum prize so far and which boxes have been opened. So, the state space is again exponential in the number of boxes. The actions are to choose boxes, which give no immediate reward, and there is a further action STOP, which pays out the final reward.

**Example 9.1.** Consider the case of two boxes. The prize in the first box is 4 with probability  $\frac{1}{2}$  and 0 otherwise. The prize in the second box is 2 with probability 1. Each box costs 1 to open.

The optimal policy in this case is to first open the first box. If we find the prize of 4, there is no point in opening the second box; our reward is  $4 - 1 = 3$ . If we do not find the prize, we open the second box; our reward is  $2 - 2 = 0$ . So, the expected reward is  $\frac{3}{2}$ .

There are multiple other policies. For example, we could open the second box first. Regardless of what we do then, the expected reward will always be 1.

The problem was introduced by Weitzman in 1979. In his original paper, he calls the acting agent "Pandora". If you are familiar with Greek mythology, you may or may not find this appropriate. One aspect is true for sure: We might regret having opened a box. If we find a better prize later, the cost for opening the earlier box has already been paid but its content is worthless.

## 2 The Problem of a Single Box

Let us first consider the problem in which there is only a single box. Would we open it? There is certainly no point in opening it if  $c_i > \mathbf{E}[v_i]$  because the expected prize cannot compensate the cost. If there are multiple boxes, this would only be worse. Therefore, we assume without loss of generality that  $c_i \leq \mathbf{E}[v_i]$  for all  $i$ . We simply ignore the boxes for which this does not hold.

Now, suppose there is an investor offering us a deal: They cover the cost of opening the box but keep some of the prize in return. For precisely, they will open the box and keep everything of the prize above a *cap*  $\sigma_i$  (to be defined below). We keep only everything below the cap.

That is, we split the value  $v_i$  into two parts, namely a *capped value*  $\kappa_i$  and a *bonus*  $b_i$ : If  $v_i \leq \sigma_i$ , then  $\kappa_i = v_i$  and  $b_i = 0$ . Otherwise, if  $v_i > \sigma_i$ , then  $\kappa_i = \sigma_i$  and  $b_i = v_i - \sigma_i$ . Equivalently, we can set  $b_i = \max\{0, v_i - \sigma_i\}$ ,  $\kappa_i = v_i - b_i$ . By these definitions always  $\kappa_i + b_i = v_i$  and  $\kappa_i \leq \sigma_i$ . So, in other words, we cap the value  $v_i$  at  $\sigma_i$ . Everything above the cap is moved to  $b_i$ .

**Example 9.2.** Consider again the first box from the previous example. That is,  $v_1 = 4$  with probability  $\frac{1}{2}$ ,  $v_1 = 0$  otherwise. If  $\sigma_1 \geq 4$ , always  $\kappa_1 = v_1$ .

If  $\sigma_1 < 4$ , then  $\kappa_1 = \sigma_1$  and  $b_1 = 4 - \sigma_1$  whenever  $v_1 = 4$ . Both are 0 when  $v_1 = 0$ . So, the expected capped value is  $\mathbf{E}[\kappa_1] = \frac{1}{2}\sigma_1$ , the expected bonus is  $\mathbf{E}[b_1] = \frac{1}{2}(4 - \sigma_1) = 2 - \frac{1}{2}\sigma_1$ .

Depending on the cap  $\sigma_i$ , this may or may not be a good deal for the investor. More precisely, the investor's utility after deducting the cost will be  $\mathbf{E}[b_i] - c_i$  in expectation. We will choose  $\sigma_i$  so that this is exactly 0. We call this the *fair cap*.

To make this formal, note that for  $\sigma_i = 0$ , we always have  $b_i = v_i$  and so  $\mathbf{E}[b_i] = \mathbf{E}[v_i] \geq c_i$ . For  $\sigma_i \rightarrow \infty$ , we always have  $b_i = 0$ , meaning that also  $\mathbf{E}[b_i] = 0$ . As  $\mathbf{E}[b_i]$  is continuous in  $\sigma_i$ , there has to be a value  $\sigma_i$  for which  $c_i = \mathbf{E}[b_i]$ .

**Example 9.3.** In our example from above, the fair cap is 2 because then  $\mathbf{E}[b_1] = c_1 = 1$ .

### 3 Policy for Multiple Boxes

We can now state our fair-cap policy: Open the boxes by decreasing fair cap  $\sigma_i$ . Stop when the largest observed value  $v_{i^*}$  exceeds the highest remaining cap and select  $i^*$ .

That is, we can without loss of generality assume that the boxes are ordered such that  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ . We then open the boxes in a fixed order  $1, 2, \dots, n$  until at some point  $\sigma_i < \max_{i' < i} v_{i'}$ , at which point we stop.

One way to think about this policy is as follows. The fair cap  $\sigma_i$  expresses what prize we can hope to get from box  $i$  after having deducted the cost. We start with the most promising box and continue opening boxes up to the point at which we do not hope to gain anything from opening any of the remaining boxes.

One interesting aspect of the policy is that the order in which boxes are opened does not depend on the observations. We will see a related but different problem very soon, in which one has to adapt choices.

**Theorem 9.4.** The fair-cap policy is optimal.

In the following, we will use two kinds of indicator random variables to denote the choices by a policy.

- Let  $I_i = 1$  if the policy opens box  $i$ . (It *inspects* the box.)
- Let  $A_i = 1$  if the policy keeps the prize in box  $i$ . (It *accepts* the box.)

First, we will express the expected reward of any policy in terms of its  $I_i$  and  $A_i$  random variables.

**Lemma 9.5.** The expected value of any policy  $\pi$  is given by

$$V(\pi) = \sum_i \mathbf{E}[A_i \kappa_i - (I_i - A_i)b_i] .$$

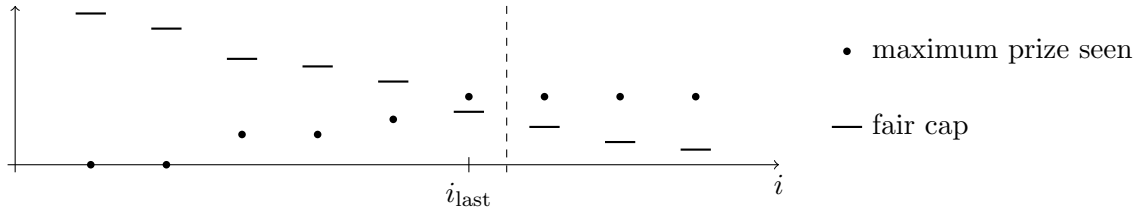


Figure 1: The fair caps only decrease while the maximum prize seen so far only increases.

So, in words, in expectation, the reward is equal to the capped value of the box whose prize is accepted minus the bonuses of all boxes that are opened but not accepted.

*Proof.* By definition

$$V(\pi) = \mathbf{E} \left[ \sum_i A_i v_i - \sum_i I_i c_i \right] .$$

By definition,  $v_i = \kappa_i + b_i$ . So, by linearity of expectation, we have

$$\mathbf{E} \left[ \sum_i A_i v_i - \sum_i I_i c_i \right] = \sum_i (\mathbf{E} [A_i v_i] - \mathbf{E} [I_i] c_i) = \sum_i (\mathbf{E} [A_i \kappa_i] + \mathbf{E} [A_i b_i] - \mathbf{E} [I_i] c_i) .$$

Now, we use the definition of  $\sigma_i$  as the fair cap. Therefore, we have  $c_i = \mathbf{E} [b_i]$ . Furthermore,  $b_i$  is a random variable that only depends on  $v_i$  whereas  $I_i$  cannot depend on  $v_i$  — when making the decision to open box  $i$ , the policy does not know  $v_i$ . Therefore  $\mathbf{E} [I_i] c_i = \mathbf{E} [I_i] \mathbf{E} [b_i] = \mathbf{E} [I_i b_i]$ . So, overall

$$V(\pi) = \sum_i (\mathbf{E} [A_i \kappa_i] + \mathbf{E} [A_i b_i] - \mathbf{E} [I_i] c_i) = \sum_i (\mathbf{E} [A_i \kappa_i] + \mathbf{E} [A_i b_i] - \mathbf{E} [I_i b_i]) .$$

The statement follows by linearity of expectation. □

In the remainder, we will show that the fair-cap policy maximizes  $\sum_i A_i \kappa_i - (I_i - A_i) b_i$  among all policies and is therefore optimal. We will do this in two steps.

**Lemma 9.6.** *The fair-cap policy always selects the box of highest capped value. That is,*

$$\sum_i A_i \kappa_i = \max_i \kappa_i .$$

For the proof, it is important that the fair caps only decrease but the maximum observed prize only increases as it is visualized in Figure 1.

*Proof.* Let  $i_{\text{last}}$  be the index of the last box to be opened and let  $i^* \leq i_{\text{last}}$  be the index of the box that we accept. We would like to show that  $\kappa_i \leq \kappa_{i^*}$  for all  $i$ . To this end, we distinguish whether the prize that we accept exceeds its cap or not.

**Case 1:**  $v_{i^*} \leq \sigma_{i^*}$ . So  $\kappa_{i^*} = v_{i^*}$ .

For  $i \leq i_{\text{last}}$ , we have

$$\begin{aligned} \kappa_i &\leq v_i && \text{(by definition)} \\ &\leq v_{i^*} && \text{(because } v_{i^*} \text{ is the highest of all prizes up to } i_{\text{last}}) \\ &= \kappa_{i^*} && \text{(because } v_{i^*} \leq \sigma_{i^*}) \end{aligned}$$

For  $i > i_{\text{last}}$ , we have

$$\begin{aligned}
 \kappa_i &\leq \sigma_i && \text{(by definition)} \\
 &\leq \sigma_{i_{\text{last}}+1} && \text{(by monotonicity)} \\
 &\leq v_{i^*} && \text{(because we stop opening boxes)} \\
 &= \kappa_{i^*} && \text{(because } v_{i^*} \leq \sigma_{i^*}\text{)}
 \end{aligned}$$

**Case 2:**  $v_{i^*} > \sigma_{i^*}$ . So  $\kappa_{i^*} = \sigma_{i^*}$ .

In this case,  $i^* = i_{\text{last}}$  because  $\sigma_{i^*+1} \leq \sigma_{i^*}$ , so we do not open box  $i^* + 1$ .

For  $i < i_{\text{last}}$ , we have

$$\begin{aligned}
 \kappa_i &\leq v_i && \text{(by definition)} \\
 &\leq \sigma_{i_{\text{last}}} && \text{(because we did not stop opening boxes)} \\
 &= \kappa_{i^*} && \text{(as observed)}
 \end{aligned}$$

For  $i > i_{\text{last}}$ , we have

$$\begin{aligned}
 \kappa_i &\leq \sigma_i && \text{(by definition)} \\
 &\leq \sigma_{i_{\text{last}}} && \text{(by monotonicity)} \\
 &= \kappa_{i^*} && \text{(as observed)}
 \end{aligned}$$

□

**Lemma 9.7.** *Our policy always fulfills  $(I_i - A_i)b_i = 0$  for all  $i$ .*

*Proof.* If  $I_i = 0$  or  $b_i = 0$ , the statement follows trivially. So, we only have to understand what happens if  $I_i = 1$  and  $b_i > 0$ . Consider the situation that the policy opens a box and  $b_i > 0$ . In this case,  $v_i > \sigma_i$ . So, it is certainly the last box to be opened. Furthermore, because box  $i$  was opened, the maximum value found in boxes  $1, \dots, i-1$  is at most  $\sigma_i$ . That is,  $v_i$  is the highest value found in boxes  $1, \dots, i$  and therefore  $A_i = 1$ . □

*Proof of Theorem 9.4.* Consider any other policy  $\pi'$  and let its indicators be denoted by  $(A'_i)_{i \in [n]}$ ,  $(I'_i)_{i \in [n]}$ . By Lemma 9.5, we have

$$V(\pi') = \sum_i \mathbf{E} [A'_i \kappa_i - (I'_i - A'_i)b_i] \leq \sum_i \mathbf{E} [A'_i \kappa_i] \leq \mathbf{E} \left[ \max_i \kappa_i \right].$$

For our policy, we have  $(I_i - A_i)b_i = 0$ . So,

$$V(\pi) = \sum_i \mathbf{E} [A_i \kappa_i - (I_i - A_i)b_i] = \mathbf{E} \left[ \sum_i A_i \kappa_i \right] = \mathbf{E} \left[ \max_i \kappa_i \right].$$

□

## References

- Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*. 1979.
- Blog post by Bo Waggoner: <http://www.bowaggoner.com/blog/2018/07-20-pandoras-box/>