Buy or Rent

Hsiang-Hsuan (Alison) Liu

Consider that you are in a ski resort and want to have a wonderful ski holiday. You do not have a ski yet, so every time when you want to ski, you have to rent one and pay the renting fee. Another option is, you can buy a pair of ski and pay for the one-time (large) fee for purchasing.

You want to minimize the cost spending on the ski. Assume that the buying price is B and the renting price is 1 (with some normalization). In an offline setting, you know how many days it is fine for skiing. Apparently, if there will be at least B days for skiing, the best strategy is to buy the ski at the first day.

However, in the real-world scenario, you do not know how the weather would be in the coming days. That is, even if you book for the hotel for more than B days, it could be the case that you got a storm and stuck in the hotel for most of the days. In this case, buying the ski on the first day is not cost-efficient for you. Hence, we have the following online problem on when to buy the ski:

Buy-or-Rent problem

Assume that the price for buying a pair of ski is B and the renting price is 1 per day. At which day of skiing should you buy a ski?

As this problem is naturally an online problem, our aim is minimizing the competitive ratio. That is, we hope to minimize the ratio between the cost incurred by our algorithm to the offline optimal cost under the worst case.

1 Two simple strategies

First let us look at two simple strategies:

- **Strategy** 1: Never buy the ski; whenever you want to ski, rent it and pay the fee 1.
- **Strategy** 2: Buy the ski on the first skiing day and pay for the fee B.

How good or how bad these strategies are? It depends on how many days you ski. For Strategy 1, the worst case is you can ski for a very long vacation, say, for d days, where $d >> B$. In this case, the optimal strategy is to buy the ski on the first day and pay for price B, while anyone who follows Strategy 1 pays for price d. The competitive ratio of this strategy is $\frac{d}{B}$, which is infinitely large as $d >> B$.

For the Strategy 2, the worst case happens when there is a crazy storm on the second day and destroys the whole ski resort completely. In this case, the optimal strategy would be renting the ski once. The competitive ratio of this strategy is $\frac{B}{1} = B$, which is incompatible as we consider B is large.

Adversary and lower bounds of an online algorithm. Recall that an adversarial input is trying to make the competitive ratio of the online algorithm as high as possible. The instance that keeps skiing is an adversary for Strategy 1, as the instance makes the decision performs as bad as possible. Similarly, the instance that stops skiing on the second day is an adversary for Strategy 2; skiing only one day is the worst thing happens when you bought the ski on the first day.

By designing an adversarial input targeting on an online algorithm, one finds a lower bound of the algorithm's performance. For example, consider Strategy 2, the instance which stops skiing on the second day shows that Strategy 2 cannot be better than B-competitive.

2 A competitive online algorithm

To have a competitive online algorithm, it is essential to decide when to buy the ski. Consider the following strategy: *keep renting the ski and buy on the B-th skiing day* (see Algorithm 1). We can prove that this algorithm is constant competitive.

Algorithm 1 A $(2 - \frac{1}{B})$-competitive online algorithm for the Buy-or-Rent problem

```
if it is the B-th day of skiing then
    Buy
else if the skiis are not bought yet
    Rent
```

Recall the attempt we mentioned in the Introduction. A trivial way to upper-bound the competitive ratio is via the following observation. If there are d days of skiing (where $d \geq 1$), the optimal strategy has to pay at least 1 for either renting or buying the ski. On the other hand, the online algorithm pays no more than $d \cdot B$ (which is an impossible worst case where the online algorithm buys the ski every day). Therefore, for any instance, the ratio $\frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)} \leq \frac{d \cdot B}{1} = dB$. That is, the competitive ratio of the online algorithm is at most dB.

The analysis is of course not ideal enough. We can improve it by some more observations on the online algorithms. For example, one can observe that the online algorithm never pays more than $2B-1$ since it buys the ski on the B-th day. Hence, we can get a better upper bound $\frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)} \leq \frac{2B-1}{1} = 2B-1$. In fact, we can further improve the analysis by some other observations on both the online algorithm and the optimal strategy:

Theorem 1. *Assume the buying price is B and the renting price is 1, the algorithm Algorithm 1 is $(2 - \frac{1}{B})$-competitive for the Buy-or-Rent problem.*

Proof. For analysis, we assume that the actual number of skiing days is d. There are two cases: $d < B$ or $d \geq B$. First, we can observe that if $d < B$, the optimal offline strategy is to rent the ski on each skiing day and the total cost of the optimal offline strategy is d. On the contrary, if $d \geq B$, the optimal offline strategy is to buy the ski on the first day and the total cost is B.

Now let us see how much the online algorithm pays in different scenarios. If $d < B$, according to the algorithm, we rent the ski every time and pay for d in total. If $d \geq B$, we buy the ski on the B-th day. In this case, we have to pay for the renting price on the first $B - 1$ days and the buying price on the B-th day. The total cost of the algorithm is $(B - 1) \cdot 1 + 1 \cdot B = 2B - 1$.

$$\begin{aligned} \text{The competitive ratio of the algorithm } &= \max\{\frac{d}{d}, \frac{2B-1}{B}\} \\ &= 2 - \frac{1}{B} \end{aligned}$$

□

Is the analysis tight? Let's see how bad the performance of this algorithm can be. Consider the adversary I that there are exactly B skiing days. In this case, the algorithm will buy the ski and right after that the ski season is over. Hence, the ratio of this adversarial instance $\frac{\mathrm{ALG}(I)}{\mathrm{OPT}(I)} = \frac{(B-1)+B}{B} = 2 - \frac{1}{B}$, which matches the upper bound of the algorithm. Therefore, the analysis is tight.

3 A general lower bound for the Buy-or-Rent problem

For the Buy-or-Rent problem, we can prove that no online algorithm performs better than $(2 - \frac{1}{B})$-competitive. We prove this by showing that for any (deterministic) online algorithm, there exists an instance such that the algorithm cost is at least $2 - \frac{1}{B}$ times of the optimal cost on the same instance.

In other words, since there is at least one instance making the ratio between the algorithm cost to the optimal cost greater than or equal to $2 - \frac{1}{B}$ for any algorithm, it is impossible for the algorithm to have a competitive ratio smaller than $2 - \frac{1}{B}$.

Theorem 2. *There is no deterministic online algorithm better than* $(2 - \frac{1}{B})$*-competitive.*

Proof. Consider any deterministic online algorithm, it must buy the ski on the k-th day for some integer $k \geq 1$ (where $k = \infty$ if the algorithm never buy the ski). For algorithm ALG_k which buys the ski on the k-th day, we design the adversarial input I_k that skies for exactly k days. With this instance, the cost of algorithm ALG_k is $(k-1)+B$, while the optimal cost is $\min\{B, k\}$.

There are two cases of k: $k \geq B$ or $k < B$. If $k \geq B$, the optimal cost is B and the ratio $\frac{\text{ALG}_k(I_k)}{\text{OPT}(I_k)} = \frac{(k-1)+B}{B} \geq \frac{(B-1)+B}{B} = 2 - \frac{1}{B}$. If $k < B$, the optimal cost is k and the ratio $\frac{\text{ALG}_k(I_k)}{\text{OPT}(I_k)} = \frac{(k-1)+B}{k}$. The ratio decreases as k increases. Hence, the ratio is lower bounded by $\frac{(B-1)+B}{B}$ since $k < B$. Therefore, for both cases, the ratio $\frac{\text{ALG}_k(I_k)}{\text{OPT}(I_k)} \geq 2 - \frac{1}{B}$.

□

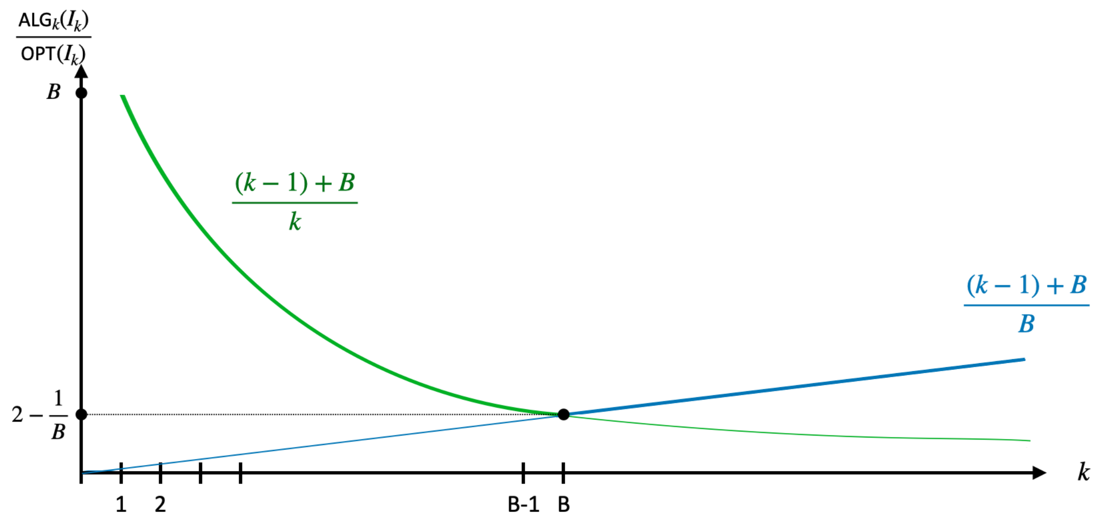

Figure 1: Illustration of the Ski-Rental problem lower bound. For any algorithm ALG_k which buys the ski on the k-th day, there exists an instance where $d = k$ that makes the ratio between the algorithm cost and the optimal cost at least $2 - \frac{1}{B}$.

By Theorem 1 and Theorem 2, we have the following corollary:

Corollary 1. *Algorithm 1 is an optimal online algorithm for the* Buy-or-Rent *problem.*