Exercise XIII, Algorithms 2024-2025

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. There are many problems on this set, solve as many as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

1 Online Algorithms

1 Imagine the cost to buy skis (B) is 120 CHF and the cost to rent skis (R) is 30 CHF per season. Consider the "Deterministic Break-Even" algorithm: rent for $B/R - 1$ seasons, then buy the skis. (So, rent for $120/30 - 1 = 3$ seasons, then buy).

(a) If you go skiing for exactly 2 seasons, what is the cost of this algorithm? What would have been the optimal cost?

(b) If you go skiing for exactly 4 seasons, what is the cost of this algorithm? What would have been the optimal cost?

(c) If you go skiing for 6 seasons, what is the cost of this algorithm? What would have been the optimal cost?

2 Consider the Rent or Buy (Ski Rental) algorithm that we saw in class, where we assume renting costs one unit, i.e., $R = 1$. Suppose you modify the rule to "rent until you've paid $c \cdot B$, then buy." Express the competitive ratio as a function of c.

3 Prove that no deterministic online algorithm for the ski-rental problem can achieve a competitive ratio strictly less than 2 when the cost of buying $B \to \infty$ (i.e. show 2 is optimal) by constructing an adversarial number of ski trips.

4 Consider a cache with size $k = 3$ (meaning it can hold 3 pages). Initially, the cache is empty. Process the following sequence of page requests: A, B, C, D, A, B, E, D, A, B, C

(a) For the LRU (Least Recently Used) algorithm, list the cache contents after each request and count the total number of cache misses. A cache miss occurs if the requested page is not in the cache. If a miss occurs and the cache is full, LRU evicts the page that has been unused for the longest time.

(b) For the FIFO (First-In, First-Out) algorithm, list the cache contents after each request and count the total number of cache misses. If a miss occurs and the cache is full, FIFO evicts the page that has been in the cache the longest (like a queue).

(c) Based on your results for this sequence, which algorithm performed better?

5 Imagine you are a hiring manager tasked with selecting the single best candidate for a job out of n total applicants. The hiring process has a peculiar structure:

1. **Group A (The Random Sample):** From the total pool of n applicants, a group of $n/2$ candidates is **chosen uniformly at random** to form Group A. These $n/2$ candidates arrive first, one by one, in a **uniformly random permutation** (i.e., their internal order within Group A is random).
2. **Group B (The Adversary's Play):** The remaining $n/2$ candidates (those not chosen for Group A) form Group B. These candidates also arrive one by one, but their arrival order is determined by an **adversary** who knows your hiring strategy for this second group and wants to prevent you from hiring the overall best candidate.

As in the classic problem:

- When a candidate arrives, you learn their "score" or rank relative to those already seen.
- You must decide immediately whether to hire them or reject them.
- The decision is **irrevocable**. If you hire someone, the process stops. If you reject them, you cannot hire them later.
- You can hire at most one candidate.
- Your goal is to maximize the probability of hiring the **single overall best candidate** from the entire pool of n applicants.

Your Task: Show that you can devise a strategy that ensures you hire the overall best candidate with a probability of at least $1/4$, regardless of the adversary's actions in Group B (for $n \geq 2$).

2 Weighted Majority and Hedge

6 Recall from the lecture that the number of mistakes that Weighted Majority makes is at most $2(1+\epsilon) \cdot (\#\text{ of } i\text{'s mistakes}) + O(\log N/\epsilon)$, where i is any expert and N is the number of experts.

Give an example that shows that the factor 2 is tight in the above bound. The simplest such example only uses two experts, i.e., $N = 2$, and each of the experts is wrong roughly half of the time. Finally, note how your example motivates the use of a random strategy (as in the Hedge strategy).

7 In this exercise we consider the Hedge algorithm and use the same notation as in the lecture notes. The average [external] regret of Hedge is defined as

$$\frac{\sum_{t \leq T} \vec{p}^{(t)} \cdot \vec{m}^{(t)} - \min_i \sum_{t \leq T} m_i^{(t)}}{T}$$

i.e., how much we "regret", on average over the days, compared to the best single strategy i.

7a If you knew the number of days T in advance, how would you set the parameter ϵ of Hedge to minimize the average external regret?

7b *(*)* Even if you do not know T in advance, describe a strategy that achieves roughly the same average external regret as in the case when T is known.

Hint: There is no need to redo the analysis from scratch. For example, you could consider restarting the algorithm each time you get to a day t of the form 4^i.