Exercise 8: Problem Lower Bounds

1 Ever Given

Recall the Ever Given problem, where waiting until the canal is available takes F units of time, and going around Africa via the Cape of Good Hope takes S. Prove that for this problem, there is no deterministic online algorithm better than $(2 - \frac{F}{S})$-competitive.

Proof. Consider any deterministic online algorithm. It must turn around to the Cape of Good Hope route at some time. T. For algorithm ALG_T which turns around at the T-th unit of time, we design the adversarial input I_T that the canal is available at time $T + \epsilon$, where $\epsilon > 0$ but very small. With this instance, the cost of algorithm ALG_T is $T + S$, while the optimal cost is $\min\{S, T + \epsilon + F\}$.

There are two cases of T: $T \geq S - F$ or $T < S - F$.

- If $T \geq S - F$, the optimal cost is S and the ratio $\frac{\text{ALG}_T(I_T)}{\text{OPT}(I_T)} = \frac{T+S}{S} \geq \frac{(S-F)+S}{S} = 2 - \frac{F}{S}$.
- If $T < S - F$, the optimal cost is less than or equal to $T + F + \epsilon$. Therefore, the ratio $\frac{\text{ALG}_T(I_T)}{\text{OPT}(I_T)} \geq \frac{T+S}{T+\epsilon+F}$. The ratio decreases as T increases. Hence, since $T < S - F$, the ratio is lower bounded by $\frac{(S-F)+S}{(S-F)+\epsilon+F} = 2 - \frac{F}{S}$ when ϵ tends to 0.

Therefore, for both cases, the ratio $\frac{\text{ALG}_T(I_T)}{\text{OPT}(I_T)} \geq 2 - \frac{F}{S}$.

□

2 Two-store Ski-Rental

Consider that there are two stores in the Ski-rental problem, 1 and 2, where you can buy or rent a pair of skis. Let r_1 and B_1 be the renting and buying prices from Store 1, respectively. For Store 2, r_2 and B_2 are defined symmetrically. Note that there is no specific relation between r_1 and r_2 or B_1 and B_2, except that B_1 and B_2 are both larger than $\max\{r_1, r_2\}$. Let $r_{\min} = \min\{r_1, r_2\}$ and $B_{\min} = \min\{B_1, B_2\}$. Show that the problem lower bound is at least $2 - \frac{r_{\min}}{B_{\min}}$.

(Hint: You can first show that the algorithms that always rent from the store with the lower renting price and buy from the store with a lower buying price must have a better ratio. Then, you focus on the algorithms in this family and find adversarial instances for any of them.)

Let S_r be the store with the renting price of $r_{\min}$, and S_B be the store with the buying price of $B_{\min}$. Let $\mathcal{A}$ be a family of algorithms that always rent the ski from S_r and buy the ski from S_B. We first show that any algorithm ALG can be transformed into an algorithm ALG$\mathcal{A}$ without increasing its cost on any instance. That is, for any ALG, there exists $\text{ALG}' \in \mathcal{A}$ such that $\text{ALG}(\sigma) \geq \text{ALG}'(\sigma)$ for any instance $\sigma = (r_1, B_1, r_2, B_2, d)$, where d is the number of skiing days. If this claim is true, then we only need to find adversarial instances for algorithms in $\mathcal{A}$.

The transformation from ALG to $\text{ALG}' \in \mathcal{A}$ is as follows. If ALG always buys the ski from S_B and rents the ski from S_r, $\text{ALG} \in \mathcal{A}$. (That is, we can set $\text{ALG}' = \text{ALG}$.) Otherwise, we set that ALG' buys the ski on the same day when ALG buys the ski but from S_B. Similarly, if there is a day when ALG rents the ski from the store $3 - S_r$,[†] ALG' rents the ski from the store S_r on the same day. Since the cost of renting/buying for ALG' is always lower than or equal to the cost paid by ALG, $\text{ALG}(\sigma) \geq \text{ALG}'(\sigma)$ for any instance σ.

[†]It's an unnecessary but fancy way to say that if $S_r = 1$, the other store is 2, and if $S_r = 2$, then the other store is 1.

Now, for any algorithm ALG that buys the ski on the k-th day, we set the adversary instance $\sigma' = (r_1, B_1, r_2, B_2, k)$. That is, the adversary has k skiing days. In this case, $\text{ALG}(\sigma) \geq \text{ALG}'(\sigma) = (k-1) \cdot r_{\min} + B_{\min}$, where ALG' is the corresponding algorithm from the transformation mentioned above. On the other hand, $\text{OPT}(\sigma') = \min\{k \cdot r_{\min}, B_{\min}\}$. Therefore, for any ALG, $\frac{\text{ALG}(\sigma')}{\text{OPT}(\sigma')} \geq \frac{\text{ALG}'(\sigma')}{\text{OPT}(\sigma')} \geq \frac{(k-1)\cdot r_{\min} + B_{\min}}{\min\{k \cdot r_{\min}, B_{\min}\}} \geq \max\{\frac{(k-1)\cdot r_{\min}+B_{\min}}{k \cdot r_{\min}}, \frac{(k-1)\cdot r_{\min}+B_{\min}}{B_{\min}}\}$. Letting $k \cdot r_{\min} = B_{\min}$ gives an lower bound of $\max\{\frac{(k-1)\cdot r_{\min}+B_{\min}}{k \cdot r_{\min}}, \frac{(k-1)\cdot r_{\min}+B_{\min}}{B_{\min}}\} = \frac{B_{\min} - r_{\min} + B_{\min}}{B_{\min}} = 2 - \frac{r_{\min}}{B_{\min}}$.[‡]

3 Online Load Balancing on Two Machines

Recall the online load balancing problem, where there are m machines. A sequence of n jobs arrives where each job J_i has a processing load of ℓ_i. The goal is to assign the jobs so that the highest load of the machines is minimized. Show that when $m = 2$, no online algorithm can be better than 1.5-competitive.

Given any algorithm ALG, the adversary σ first releases two unit-size jobs. The input sequence stops if ALG assigns the two jobs on the same machine. In this case, OPT cost is 1 while ALG cost is 2, and the ratio $\frac{\text{ALG}(\sigma)}{\text{OPT}(\sigma)} = 2 > 1.5$.

On the other hand, if ALG puts these two jobs on different machines, the adversary releases another job with size 2. In this case, the OPT cost is 2 by putting the size-2 job on one machine and the other two jobs on the other machine. However, ALG is 3, and the ratio between ALG and OPT is $\frac{3}{2}$.

That is, in either case, the ratio between the ALG cost and the OPT cost is at least 1.5.

4 Potential function of Ski-Rental

Recall the Ski-Rental problem and the algorithm that buys the ski on the B-th skiing day. Use the potential function method to show that the algorithm is 2-competitive.

(Hint: In this problem, the parameters observable from outside are: if the algorithm/optimal have a pair of ski, and how many skiing days are there so far.)

Let $A \in \{0, 1\}$ (and $B \in \{0, 1\}$) indicate whether the algorithm (and the optimal solution) has a ski (1) or not (0). We define the potential function $\Phi_k(A, B)$ at the end of the k-th day:

- $\Phi_k(0, 0) = k$,
- $\Phi_k(0, 1) = 2B - k$,
- $\Phi_k(1, 0) = 0$, and
- $\Phi_k(1, 1) = 0$.

There are three cases regarding the behavior of the algorithm and the optimal solution on each day k:

1. Both the algorithm and optimal solution rent the ski: In this case, neither of the algorithms has a pair of skis. The change of potential is $\Phi_k(0, 0) - \Phi_{k-1}(0, 0) = k - (k-1) = 1$. Thus, $\text{ALG}_k + \Delta\Phi_k = 1 + 1 = 2 \leq r \cdot \text{OPT}_k = r \cdot 1$ for some r.
2. The Optimal solution is to buy a ski (no matter when it did this), and the algorithm rents. In this case, $\Phi_k(0, 1) - \Phi_{k-1}(0, 1) = (2B - k) - (2B - (k-1)) = -1$. Thus, $\text{ALG}_k + \Delta\Phi_k = 1 - 1 = 0 \leq r \cdot \text{OPT}_k$ for some r.
3. The algorithm buys a ski. If the optimal solution does not buy the ski, $\Phi_k(1, 0) - \Phi_{k-1}(0, 0) = 0 - k = -k$. Otherwise, if the optimal solution buys the ski, $\Phi_k(1, 1) - \Phi_{k-1}(0, 1) = 0 - (2B - k) = k - 2B$. Since this must happen on the B-th day, $k = B$. Thus, $\text{ALG}_k + \Delta\Phi_k \leq B + \max\{-k, k - 2B\} = B - B = 0 \leq r \cdot \text{OPT}_k = r \cdot 0$ for some r.

Picking $r = 2$ is sufficient for all the cases. Hence, the algorithm is 2-competitive.

[‡] It is consistent with the original case, which can be seen as both stores have a renting price of 1 and a buying price of B.