

Advanced Algorithms

Mo Problems

November 24, 2025

Instructions

I have made a few bonus problems that can be turned for extra credit on your assignment grade. These problems are totally optional and I **guarantee that your grade will not be hurt if you don't do them**. But if you want to do some more interesting problems or a few extra points, they are here. You can send me your solutions by email any time before December 12th. As usual, looking up answers or using AI tools is not allowed and I may ask you to explain your answer to me orally. By the way, I think problem 1 is probably the trickiest of these.

Problem 1: Optimal Scheduling (8 points)

Recall the Least-Loaded-Machine-First algorithm. We showed in class that this is a 2-approximation for the load balancing problem (see the posted notes on job scheduling or Section 11.1 of KT). I also mentioned in class that if you perform this same algorithm but first sort the jobs in non-increasing order of size, then the algorithm is actually a $4/3$ -approximation.

Is it true that for every instance of the load balancing problem, there exists an order of the jobs so that when Least-Loaded-Machine-First processes the jobs in this order, it produces an optimal solution? Decide whether you think this is true or false, and give either a proof or a counterexample.

Solution.

Problem 2: Additive Traveling Salesperson (6 points)

We have mainly worked with approximation guarantees in a multiplicative sense. But sometimes an additive guarantee is also reasonable. We say that an algorithm is an **additive** α -approximation if the solution produced by the algorithm is always at most $OPT + \alpha$ on every instance.

For example, in the Metric Traveling Salesperson Problem, an additive 100-approximation is an algorithm which always returns a Hamiltonian cycle with cost at most 100 more than the cheapest one. Show that there is no polynomial-time additive 100-approximation algorithm for metric TSP unless $P = NP$.

Solution.

Problem 3: Learning Augmented Ski-Rental (10 points)

Recall the ski-rental problem. We may buy a skis for b dollars and rent daily for r dollars per day. For simplicity, we will assume that the price of renting r divides the price of buying b . Once we have made this assumption, we can further assume without loss of generality that the price of renting is 1 and the price to buy skis is $B := b/r$. So for the remainder of this problem, the renting price is 1 and the buying price is some integer B .

In the standard ski rental problem, we saw that the Better-Late-Than-Never algorithm always pays at most 2 times the cost of the clairvoyant optimal solution on any input. This is an amazing guarantee, but it may seem unreasonable in our model that we have absolutely no knowledge of when the ski-season will end. In practice, we might have plenty of historical data, as well as the current time of year, recent weather patterns and various information which will be features in a Machine Learning model which outputs a prediction $\hat{D}$ of the day the ski-season will end.

Ski-rental with predictions of unknown quality: Suppose that there is a black-box that predicts the number of days of the skiing season as $\hat{D}$ while the true length of the skiing season is D days. The error of this prediction is defined as

$$\eta = |\hat{D} - D|,$$

which, importantly, is unknown to us.

Consider the algorithm known as Follow The Prediction (FTP). In this algorithm, if $\hat{D} \geq B$, we buy the skis on day 1. If $\hat{D} < B$, we rent skis every day until the season is over. As usual, OPT is defined as $\min\{B, D\}$ which is the cost of the optimal clairvoyant algorithm which knows when the season will end.

- (a) Consider a ski-rental instance with $B = 10$. Suppose the prediction says that the season will last 6 days, so $\hat{D} = 6$. However, suppose that the season actually lasts 15 days, so $D = 15$. What will the algorithm do on this input? What will its cost be and what is the optimal cost? What is the prediction error?
- (b) Do the same as part (a) but with $B = 10$, $\hat{D} = 12$, and $D = 4$. This is a case where FTP buys the skis while the optimal solution rents.
- (c) Give an example of an input (a value of B , $\hat{D}$, and D), where FTP pays exactly $OPT + \eta$.
- (d) Show that Follow the Prediction pays at most $OPT + \eta$ on every input.
Hint: consider the two cases separately where $\hat{D} \geq B$, and where $\hat{D} < B$, and show that FTP pays at most $OPT + \eta$ in either case.
- (e) Conclude that the competitive ratio of Follow The Prediction is exactly $1 + \frac{\eta}{OPT}$.

The above analysis shows that we can exploit the prediction to get an improved guarantee as long as the prediction is reasonably accurate. However, it should be noted that there are ways exploiting the prediction when it is accurate while also remaining robust if the prediction is way off. See this recent paper if you are interested: Improving Online Algorithms via ML Predictions.

Solution.