Khoury College of Computer Sciences — Fall 2023
Northeastern University — 5 December 2023
CS 7870: Seminar in Theoretical Computer Science — Scribe: Mohammad Saneian

Lecture 23 Outline:

- The online primal-dual paradigm
- Ski rental
- Set cover

In this lecture, we cover two problems using the online primal-dual approach. At a high level, we first translate our problem into a linear program. Then, at each instant, we maintain two feasible LP solutions: one for the primal (which will be the output of the online algorithm) and one for the dual. The competitive ratio will be the ratio between the primal cost and the dual cost. The online primal-dual paradigm has been used for a dozens of online algorithmic problems; it has been formalized in the pioneering work by Buchbinder and Naor [BN09]. The material in this class is largely based on the preceding research monograph.

1 Ski rental

In a previous lecture, we presented optimal deterministic and randomized algorithms for ski rental. Here we present the randomized algorithm using the primal-dual paradigm; recall that the algorithm achieves a better competitive ratio of $\frac{e}{e-1}$.

Recall the ski rental problem. A person is going skiing for an unknown number of days. Renting skis costs 1 per day and buying skis costs B. Every day, the person must decide whether to continue renting skis for one more day or buy a pair of skis. If the person knows in advance how many days she will go skiing, she can decide her minimum cost. If she will be skiing for more than B days it will be cheaper to buy skis but if she will be skiing for fewer than B days it will be cheaper to rent. What should she do when she does not know in advance how many days T she will ski?

We can define a set of variables for a linear program for how an algorithm makes the choices of buying or renting at each day t. Let us define x to be the indicator that we have bought the ski and z_t to be the indicator that we rent on day t. Then we can form the primal-LP as follows:

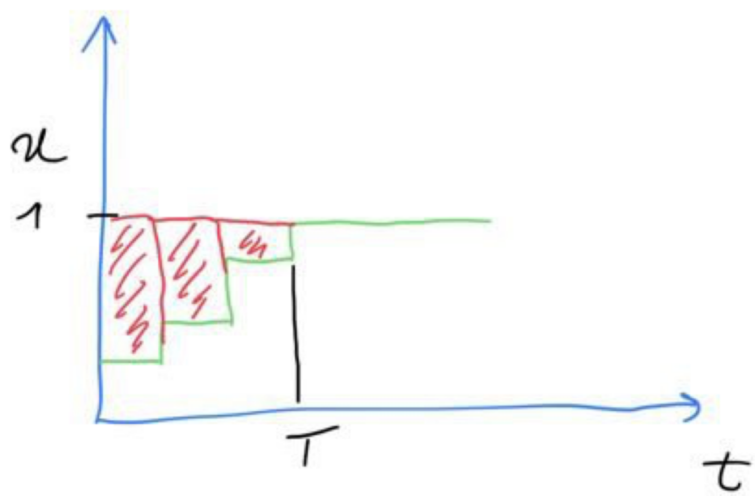
$$\begin{array}{rrcl} \min & B \cdot x + \sum_t z_t & & \\ \text{s.t.} & \forall t : x + z_t & \geq & 1 \\ & x, z_t & \geq & 0 \\ & x \text{ Monotonic} & & \end{array}$$

The objective function sums z_t over all days which is the cost of all the days we rent and $B \cdot x_t$ which will be B if we bought ski in the first T days. The first set of constraints is for being able to ski on the t-th day (for all t) and monotonicity of x means that when we buy once we have the ski for the rest of the days. We can write a dual program from this primal LP which will be the following:

$$\begin{array}{rrcl} \max & \sum_t y_t & & \\ \text{s.t.} & \sum_t y_t & \leq & B \\ & y_t & \leq & 1 \\ & y_t & \geq & 0 \end{array}$$

The way the algorithm works is that we will maintain a solution to both the primal and the dual. The solution to the dual is used as a lower bound on optimal answer and we output the primal solution. This way if the solution to the primal is $\alpha \cdot$ (dual solution) it will be at $\alpha \cdot$ opt which is what we want. The approach is that we use this online primal dual method to find an α competitive solution for the online fractional problem, for a suitable α. Next, we randomly round this solution to get an online integral solution.

Now let us take a closer look at the fractional solution for this problem. In the optimal solution we have $z_t = 1 - x$ for all t; so all that an algorithm is picking is how the function x changes over time. Pictorially we have:



In the picture above the green line is the function x which is a non-decreasing function. The shaded red parts are z_t. The cost of the algorithm for instance T would be

$$\text{cost at } T = B \cdot (\text{value of green line at } T) + \text{the shaded red areas before } T$$

The solution that this algorithm gives us is clearly primal feasible since we set $z_t = 1 - x$. Showing it is dual feasible is not trivial since we do not know if we have $\sum_t y_t \leq B$. For now we focus on comparing primal cost dual cost.

When we do one iteration the change in dual cost $\Delta D = 1$ since the only thing changing is $y_t = 1$. The change for the primal cost is:

$$\Delta P = z_t + \Delta x = 1 - x + (\frac{x}{B} + \frac{1}{cB})B = 1 + \frac{1}{c}$$

Algorithm 1: Online fractional ski-rental

input: online sequence of T days (unknown)

Set $x = 0$

for $t = 1$ *to* T **do**

  if $x < 1$ **then**

    $z_t = 1 - x$

    $x = x \cdot (1 + \frac{1}{B}) + \frac{1}{cB}$

    $y_t = 1$

return $B \cdot x + \sum_{t=1}^{T} z_t$

Therefore by applying this in all steps, we can get

$$\text{Primal cost} \leq (1 + \frac{1}{c}) \text{ Dual cost}$$

If we prove that $\{y_t\}$ provides a dual feasible solution then we can prove that the primal cost is less that $(1 + \frac{1}{c}) \cdot \text{OPT}$.

Dual feasibility: To prove dual feasibility we need to bound the sum of y_t by B. Every time this sum is increasing by 1 we have $x = x \cdot (1 + \frac{1}{B}) + \frac{1}{cB}$ and at some point $x > 1$ and we will not have $y_t = 1$.

When we have $\sum_t y_t = 1$ we have $x = \frac{1}{cB}$. Next, when $\sum_t y_t = 2$ we have $x = \frac{1}{cB} + \frac{1}{cB} \cdot (1 + \frac{1}{B})$. Generally if we have $\sum_t y_t = i$, we have

$$x = \frac{(1 + 1/B)^i - 1}{c}$$

If we set $c = (1+1/B)^B - 1$, when $\sum_t y_t = B$ we will have $x > 1$ so we will not increase the dual solution objective value more than B times and we get dual feasibility.

When B is large $c \sim e - 1$ so our competitive ratio will be

$$1 + 1/c = 1 + \frac{1}{e-1} = \frac{e}{e-1}.$$

Randomized rounding: The algorithm to round the fractional solution is very elegant and simple. We first pick $\tau = [0, 1]$ uniformly at random. The algorithm buys when $x > \tau$ happens.

We now analyze the expected cost of the randomized algorithm. Since the probability of buying skis on the jth day is equal to x_j, the expected cost of buying skis is precisely $B \cdot \sum_{j=1}^{k} x_j = Bx$, which is exactly the first term in the primal objective function. The probability of renting skis on the jth day is equal to the probability of not buying skis on or before the jth day, which is $1 - \sum_{i=1}^{j} x_i$. Since

$$z_j = 1 - \sum_{i=1}^{j-1} x_i \geq 1 - \sum_{i=1}^{j} x_i,$$

we get that the probability of renting on the jth day is at most z_j, corresponding to the second term in the primal objective function. Thus, by linearity of expectation, for any number of ski days, the expected cost of the randomized algorithm is at most the cost of the fractional solution.

2 Online set cover

In this problem, we have a collection of sets S_i, and elements arrive one by one. At each moment, we have a collection of arrived elements, and we want to maintain a subset of S_i's such that they cover all these elements. An online algorithm can be described by which sets are included in the answer at each moment. We set x_i to be the indicator that S_i is in the collection of sets. And we want to minimize the number of sets we are picking. We can form the following primal LP for these constraints.

$$
\begin{array}{lrcl}
\min & \sum_i x_i & & \\
\text{s.t.} & \forall j : \sum_{i:j \in S_i} x_i & \geq & 1 \\
 & \forall i : x_i & \geq & 0 \\
 & & \geq & 0
\end{array}
$$

The second constraint is imposing that for the element j we have one of the sets that contains it in the set cover. One thing to note is that once you pick a set S_i you can not delete it in the future. This means that we have another constraint, which will be that x_i's are monotonically increasing. The dual program will be the following:

$$
\begin{array}{lrcl}
\max & \sum_j y_j & & \\
\text{s.t.} & \forall i : \sum_{j \in S_i} y_j & \leq & 1 \\
 & \forall j : y_j & \geq & 0
\end{array}
$$

We propose the following online primal-dual algorithm for set cover. The solution is clearly primal feasible

Algorithm 2: Online set cover

input: online sequence of items $j = 1, ..., m$

- $\forall i$: Set $x_i = 0$
- **for** $j = 1$ *to* m **do**
  - item j arrives
  - **while** $\sum_{i:j \in S_i} x_i < 1$ **do**
    - $\forall i : j \in S_i$
    - $x_i = 2 \cdot x_i + \frac{1}{|i:j \in S_i|}$
    - $y_j = y_j + 1$

since we increase x_i's until it satisfies the constraint in the primal program. For one iteration, the difference in the cost of primal will be $\Delta P \leq 2$. This is because while we are in the while loop we have $\sum_{i:j \in S_i} x_i < 1$ so the terms $2 \cdot x_i$ will not have a sum of greater than 1. The terms $\frac{1}{|i:j \in S_i|}$ will also have a sum of at most 1 since they are all $\frac{1}{k}$ where k is the number of sets that contain element j and there is k of this sets.

The difference in the cost of the dual will be $\Delta D = 1$ since we have $y_j = y_j + 1$ once. This will not be a dual-feasible solution since when an item arrives several times $y_j > 1$ which violates the constraint $\sum_{j \in S_i} y_j \leq 1$. To make the solution dual feasible we use the following lemma.

Lemma 1. For a specific set S_i, we have $\sum_{j \in S_i} y_j \leq \log(d)$ where $d = \max_j |\{i : j \in S_i\}|$.

Proof. Suppose the set S_i consists of elements $j_1, ..., j_k$. When the first item j_1 arrives we set $x_i = \frac{1}{d}$. When

j_2 arrives we set $x_i = 2 \cdot \frac{1}{d} + \frac{1}{d} = \frac{3}{d}$. When j_3 arrives we set $x_i = 2 \cdot \frac{3}{d} + \frac{1}{d} = \frac{7}{d}$. This is an exponential sequence for x_i's, so after $\log d$ steps, x_i exceeds one, and the corresponding y_j will never be increased. $\square$

Now, using the above lemma, we can satisfy the dual constraints with a $\log d$ factor meaning $\sum_{j \in S_i} y_j \leq \log(d)$. To make the solution dual feasible all we need to do is have $y_j = y_j + \frac{1}{\log d}$ in line 7 of algorithm 2. This will make $\Delta P = 2$ and $\Delta D = \frac{1}{d}$ so our competitive ratio will be

$$\frac{\Delta P}{\Delta D} = O(\log d)$$

References

[BN09] Niv Buchbinder and Joseph Seffi Naor. The design of competitive online algorithms via a primal–dual approach. *Foundations and Trends® in Theoretical Computer Science*, 3(2–3):93–263, 2009.