

Lecture 10 — October 2, 2014

*Prof. Jelani Nelson**Scribe: Zhengyu Wang*

In this lecture, we give more examples of primal-dual approach to design and analyze online algorithms. First, we give a roughly $\frac{e}{e-1}$ -competitive randomized algorithm for ski-rental problem in Section 1. Then we show an $O(\log m \cdot \log n)$ -competitive randomized algorithm for online set cover problem in Section 2. In the end, we talk a little bit about approximation problems, and start to analyze the greedy algorithm for offline set cover problem in Section 3.

1 Ski-rental Problem

Recap the ski-rental problem: we want to ski for indefinite days, and for each day we need to choose whether to rent ski or to buy ski. We know

- cost of renting: \$1
- cost of buying: \$ B

Our goal is to minimize total cost. Last time we saw that deterministically we could get a 2-competitive online algorithm. In this section, we show a randomized algorithm with competitive ratio $1 + \frac{1}{(1+1/B)^{B-1}}$, which is roughly $\frac{e}{e-1}$ because as B gets large, $(1 + 1/B)^B \approx e$, and the ratio goes down to $\frac{e}{e-1}$. Note that the worst case is 2 when $B = 1$, but B should be more than 1 because otherwise we will not rent skis at all. This bound is originally given by Karlin et al. [KMMO94]. But we will not show the algorithm as they did, instead we adopt a primal-dual approach to get this bound. (See Chapter 3 in [BN09], which gives this algorithm as an example.)

In the online version of primal LP, we see one constraint at a time. We have to make online decision to change the variables in order to satisfy the constraint. And through the time the variables should increase monotonously. z_i indicates whether to rent on the i -th day, and x indicates whether we buy ski. There are k days in total. For any day i , we have to either rent or buy ski.

Primal LP (a.k.a. covering LP):

$$\begin{aligned}
 & \min Bx + \sum_{i=1}^k z_i \\
 & \text{s.t. } \forall i \in [k], x + z_i \geq 1 \\
 & \quad x \geq 0, \forall i, z_i \geq 0
 \end{aligned} \tag{1}$$

Dual LP (a.k.a. packing LP):

$$\begin{aligned}
 & \max \sum_{i=1}^k y_i \\
 \text{s.t. } & \sum_{i=1}^k y_i \leq B \\
 & \forall i, 0 \leq y_i \leq 1
 \end{aligned} \tag{2}$$

1.1 Algorithm for Fractional Case

The main idea is to build up the online solution for both primal and dual. We maintain both solutions to be feasible. And we are going to show that the objective cost on the primal solution is not too much bigger than dual.

Below is the proposed algorithm, where c is a constant which we will specify later.

1. Initially x , all z_i and y_i are 0
2. When we see constraint i online
 - if $x = 1$, do nothing
 - else
 - (a) $z_i \leftarrow 1 - x$
 - (b) $x \leftarrow (1 + 1/B)x + 1/cB$
 - (c) $y_i \leftarrow 1$

Now we need to analyze the ratio between the objective functions in the primal and dual, i.e., $cost_P/cost_D$, where $cost_P$ is the value of primal objective (on x, z), and $cost_D$ is the value of dual objective (on y). We also want to ensure that the solutions x, z and y are both feasible.

In every step, either both objectives don't change or $\Delta(cost_D) = 1$ when we increase y_i by 1 at i -th iteration, and in that case the change in the primal objective is

$$\begin{aligned}
 \Delta(cost_P) &= B \cdot \Delta x + z_i \\
 &= B \cdot ((1 + 1/B)x + 1/cB - x) + (1 - x) \\
 &= (x + 1/c) + (1 - x) \\
 &= 1 + 1/c
 \end{aligned} \tag{3}$$

Therefore, the ratio $cost_P/cost_D$ is at most $1 + 1/c$.

Now we just need to make sure that x, z and y are both feasible.

- Primal feasibility. z_i is just $1 - x$, so the i -th constraint is exactly tight. And certainly we do not make anything negative. So the primal is feasible.

- Dual feasibility. Definitely all the y_i are between 0 and 1. What we need is just to make $\sum_{i=1}^k y_i \leq B$, which is equivalent to show that we have $x < 1$ for at most B times during the iterations. because once x rises up to 1 we will not do updates anymore.

Let the increment to x on the i -th day be x_i , so that $x = \sum_i x_i$. We have $x_1 = \frac{1}{cB}$, and the sequence x_i are geometric with ratio $q = 1 + 1/B$. We have

$$\begin{aligned} \sum_{i=1}^B x_i &= \frac{1}{cB} \cdot \sum_{j=0}^{B-1} q^j \\ &= \frac{1}{cB} \cdot \left(\frac{(1 + 1/B)^B - 1}{1/B} \right) \\ &= \frac{(1 + 1/B)^B - 1}{c} \end{aligned} \tag{4}$$

So we set $c = (1 + 1/B)^B - 1$.

Now we can give the competitive ratio. Because

$$\text{cost}_D \leq \text{OPT}_P \leq \text{cost}_P \leq (1 + 1/c)\text{cost}_D$$

We have

$$\text{cost}_P \leq (1 + 1/c)\text{OPT}_P$$

However, so far we have not solved the problem because we can't fractionally rent/buy on each day. The solution must be integral. We fix it by using randomization as follows.

1.2 Fixing to Integral Case

- Initialize all $x, z_i, y_i \leftarrow 0$
- Pick $\alpha \in [0, 1]$ uniformly at random
- When seeing new constraints, will update x, z_i, y_i as before (fractionally)
- We buy when $x \geq \alpha$, otherwise we rent

Now we prove that the expected cost of doing so equals to the cost of previous fractional solution.

Proof. We have

$$\mathbb{E}(\text{cost}) = \mathbb{E}(\text{\$spent buying}) + (\text{\$spent renting})$$

Where

$$\mathbb{E}(\text{\$spent buying}) = B \cdot \mathbb{P}(\text{buy}) = B \cdot x$$

And

$$\begin{aligned}
\mathbb{E}(\text{\$spent renting}) &= \sum_{i=1}^k \mathbb{P}(\text{rented on day } i) \\
&= \sum_{i=1}^k (1 - (x_1 + x_2 + \dots + x_{i-1})) \\
&= \sum_i z_i
\end{aligned} \tag{5}$$

□

2 Online Set Cover

In this section, we show an $O(\log n \cdot \log m)$ -competitive randomized algorithm for online set cover problem. This is one more example of primal-dual online algorithms. The bound is due to Alon et al. [AAA⁺09].

Problem Definition. We have a universe of elements $\{1, \dots, n\}$, and we have m sets $\mathcal{C} = \{S_1, \dots, S_m\}$, $S_i \subset [n]$. Our goal is to pick a sub-collection of the S_i so that their union covers the universe, and we want to minimize the number of sets taken. In the online setting, we know that there are m sets, but we do not know the elements they contain. And as one element comes online, we immediately know the sets which contain the element. If we have not covered the element, we have to choose a set to cover it.

Set cover is an NP-hard problem, and it is also NP-hard to get an $o(\log n)$ -approximation solution. Therefore we do not expect poly-time online algorithms with better than $O(\log n)$ approximation ratio, because otherwise we could have better offline algorithm by pretending the elements coming online.

We show LP relaxations for set cover problem below. In the primal LP, we have one variable x_S for each set $S \in \mathcal{C}$.

Primal LP:

$$\begin{aligned}
&\min \sum_{S \in \mathcal{C}} x_S \\
&\text{s.t. } \forall i \in [n], \sum_{i \in S} x_S \geq 1 \\
&\quad x \geq 0
\end{aligned} \tag{6}$$

Dual LP:

$$\begin{aligned}
&\max \sum_{i=1}^n y_i \\
&\text{s.t. } \forall S \in \mathcal{C}, \sum_{i \in S} y_i \leq 1 \\
&\quad y \geq 0
\end{aligned} \tag{7}$$

2.1 Algorithm for Fractional Case

We show the following algorithm

1. Initialize $x = (1/m, \dots, 1/m)$
2. When we see constraint i
 - (a) Suppose $i \in S_{j_1}, \dots, S_{j_r}$
 - (b) While the constraint is not satisfied, run a continuous process where x_S changes according to $\frac{dx_S(t)}{dt} = x_S(t)$ simultaneously for $S = S_{j_1}, \dots, S_{j_r}$ until the constraint is satisfied. (Note that after x_S evolves T time steps, it multiplicatively increases by e^T factor)

Let OPT denote the optimal solution (where OPT is a collection of sets in \mathcal{C} , and $|OPT|$ denotes the minimum number of sets to cover the universe). Whenever we run the evolution, x_S is increasing for some $S \in OPT$. For a particular $S \in OPT$, it is impossible to evolve for longer than $\ln m$ seconds. Hence the process lasts for at most $|OPT| \cdot \ln m$ seconds.

$$\begin{aligned} \frac{d(cost_P(t))}{dt} &= \frac{d(x_{S_{j_1}}(t)) + \dots + x_{S_{j_r}}(t)}{dt} \\ &= x_{S_{j_1}}(t) + \dots + x_{S_{j_r}}(t) \leq 1 \end{aligned} \tag{8}$$

Therefore, at the end of algorithm, $cost_P \leq OPT \cdot \log m$. Note that the catch here is that this is not an integral solution. We will fix it later.

The previous analysis adopts purely primal view. We can also analyze it by primal-dual approach. In the new algorithm, we

1. Maintain x_S as before
2. y is initially set to 0
3. As we run evolution to satisfy the i -th constraint, we also evolve y_i according to $\frac{dy_i(t)}{dt} = 1$.

Claim 1. (1) x is feasible; (2) $\frac{y}{\log m}$ is dual feasible.

Proof. We need to show $\forall S \in \mathcal{C}, \sum_{i \in S} y_i \leq \log m$, i.e., the dual constraints are only unsatisfied by $\log m$ factor.

Let's focus on a particular set S . Because whenever $\sum_{i \in S} y_i$ increases, we also simultaneously increase x_S , for every $i \in S$. We have already known that x_S can only increase for $\leq \log m$ seconds (before it becomes 1). Therefore, $\sum_{i \in S} y_i \leq \log m$. \square

Because

$$\frac{1}{\log m} \cdot cost_D(y) \leq cost\left(\frac{y}{\log m}\right) \leq OPT_P \leq cost_P(x) \leq cost_D(y)$$

We get $cost_P(D) \leq \log m \cdot OPT$.

2.2 Fixing to Integral Case

Now we are trying to get integral solution.

(1) First attempt

1. At the very beginning, for every $S \in \mathcal{C}$, pick uniformly random $\alpha_S \in [0, 1]$
2. Once $x_S \geq \alpha_S$, take S into our solution (the value of x_S still remains like before)

We can show that $\mathbb{E}(\text{integral solution}) = \text{cost}(\text{fractional solution})$.

The problem of this attempt is that the possibility of getting a feasible solution might be very small. We can fix it by boosting the possibility as follows.

(2) Fixing

At some point i in request sequence

1. For each S , we will pick $\alpha_S^{(1)}, \alpha_S^{(2)}, \dots, \alpha_S^{(r)} \in [0, 1]$ independently uniformly at random, where $r = c \log n$ and c is a constant.
2. If $x_S \geq \alpha_S^{(j)}$ for any j , then put S into our solution

The fixing multiplies the expected competitive ratio by $O(\log n)$. On the other hand, by the fact that $1 - p \leq e^{-p}$, $p \in [0, 1]$, we can prove that for any specific constraint, the chance that the solution does not satisfy it is $O(n^{-c})$, and by union bound we can prove that the solution is feasible with high possibility.

3 Approximation Algorithms

Now we are going to move on to approximation algorithms, which is a new but related topic. The basic gist is that:

1. We need to solve some optimization problems, for example, (offline) set cover problem
2. The best known algorithms are slow (maybe the problems are NP-hard)
3. We want a fast algorithm which produces a solution with cost $\leq \alpha \cdot OPT$ where α is relatively small (for minimization problems)

See the books [WS11, Vaz01] for more information about approximation algorithms.

Example (Set Cover Problem). There are n items and m sets. Each set S has cost c_S . We want to minimize $\sum_{S \text{ chosen}} c_S$ s.t. the chosen sets cover $\{1, \dots, n\}$.

The Algorithm. We will show an algorithm and analysis due to [Chv79, Lov75]. If all the c_S are 1, the familiar greedy algorithm is: at every step, as long as there is an element that has not been covered yet, we choose the set which covers the most of the uncovered elements

so far. There is a straight-forward corresponding algorithm for non-uniform costs, which is, as long as there exists uncovered element, put S into our solution s.t. S minimizes $\frac{c_S}{n_S}$, where n_S is the number of new elements S covers.

Now we formulate the problem in primal and dual LP in order to analyze its performance.

Primal LP:

$$\begin{aligned} \min \quad & \sum_S c_S \cdot x_S \\ \text{s.t.} \quad & \forall i \in [n], \sum_{S \ni i} x_S \geq 1 \\ & x \geq 0 \end{aligned} \tag{9}$$

Dual LP:

$$\begin{aligned} \max \quad & \sum_{i=1}^n y_i \\ \text{s.t.} \quad & \forall S \in \mathcal{C}, \sum_{i \in S} y_i \leq c_S \\ & y \geq 0 \end{aligned} \tag{10}$$

Primal-Dual Analysis:

- Initially $x, y = 0$
- When we take some set S ,
 - Set $x_S = 1$
 - For each newly covered $i \in S$, set $y_i = \frac{c_S}{n_S}$, where n_S is the number of newly covered elements at this point.

Note that the primal and dual costs are equal. Now the primal is obviously feasible, but the dual would not be feasible. We are going to see next time, the dual constraint would be violated by at most $\log n$ multiplicative factor. So after scaling down, we will get a feasible solution. This approach of scaling down to deal with dual feasibility violation is called “dual fitting” [JMM⁺03].

References

- [AAA⁺09] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009.
- [BN09] Niv Buchbinder and Joseph 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.

- [Chv79] Vasek Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4:233–235, 1979.
- [JMM⁺03] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50(6):795–824, 2003.
- [KMMO94] Anna R Karlin, Mark S Manasse, Lyle A McGeoch, and Susan Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.
- [Lov75] László Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math.*, 13:383–390, 1975.
- [Vaz01] Vijay V Vazirani. *Approximation algorithms*. springer, 2001.
- [WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.