CMSC 858F: Algorithmic Game Theory
Fall 2010
Frugality (Cont.), Adwords Auctions

Instructor: Mohammad T. Hajiaghayi
Scribe: Ioana Oriana Bercea

October 6, 2010

1 Overview

In this lecture, we are going to finish the material we covered in last week's lecture. Specifically, we are going to discuss hardness of approximation for the algorithm we presented and illustrate some interesting related problems in Computer Science. We are also going to present an approximation algorithm for profit maximization in Adwords auctions.

2 Profit Maximization and Frugality for Auctions(continued)

2.1 Scenario

First, let's review the scenario we talked about last class.We are given:

- a set of *items* labeled from 1 to m
- a set of *agents*(i.e. single minded bidders) labeled 1 to n
- each agent has a value v_i for the set S_i

Our *goal* is to price the items individually such that we maximize our profit. Our *assumptions* are that the agents can either buy or not buy a particular item(i.e. no fractional items allowed), without having the possibility to go to another store and that items are available in unlimited supply.

2.2 Algorithm

- consider the same prices for all the items
- the candidate prices are $q_i = v_i/|S_i|$

2.3 Hardness of Approximation

We have seen in the previous lecture that the analysis of the algorithm is tight. However, it is a rather simple mechanism and a natural question arises: *Can we do better?* As we shall see in the next section, the answer is essentially *No.* The way we are going to prove this is by doing a reduction from our current problem to a problem we already know is hard to approximate(Unique Coverage Problem).

3 Unique Coverage Problem

3.1 Scenario

We are given:

- a universe U of n elements
- a collection S of subsets of U

The *goal* is to find a subcollection $S' \subseteq S$ which maximizes the number of elements that are *uniquely covered.* (i.e. appear in exactly one set of S')

3.2 Hardness of Approximation

There is a simple $O(\log n)$-approximation algorithm that is similar to our previous one for the pricing problem. Surprisingly, this is as essentially the best approximation we can get. Specifically, we have the following:

Theorem 1 *The Unique Coverage Problem is hard to approximate within a factor better than $O(\log^c(n)))$, $0 < c < 1$, unless NP has a sub-exponential algorithm. Indeed, the problem is $O(\log^{1/3}(n))$ hard or even $O(\log(n))$ hard under stronger but plausible complexity assumptions.[?]*

We are not going to discuss the proof of the theorem. Rather, we are going to proceed to the aforementioned reduction.

3.3 Reduction

As we mentioned before, we are going to do a reduction from our previous pricing problem, therefore proving that the Unique Coverage Problem is a special case of our pricing problem.

We do this by doing the following mapping:

- each set S_i maps to an item I_i
- each element e_i of the universe U maps to a bidder b_i
- buyer b_i has valuation 1 for one bundle B_i, namely the set of items I_j that corresponds to sets S_j containing the e_i

Under this mapping, we get that the zero-one pricing of items corresponds to the Unique Coverage Problem. On the other hand, we can obtain the general fractional case from the zero-one case by adding some constant factor approximation. Therefore, our reduction is complete and hardness of approximation established.

4 Two Well-known Special Cases of the Pricing Problem

4.1 The Highway Problem

- n items corresponding to highway segments, labeled 1 to n
- the bidders correspond to drivers who desire bundles of highway segments represented by the interval $[i, i']$

With lots of effort, one can surpass the general case $\mathcal{O}(\log n)$-approximation and obtain a $\mathcal{O}(\frac{\log n}{\log \log n})$-approximation.[?] Recently, a PTAS with $(1+\epsilon)$-approximation has been developed.[?] However, the problem is strongly NP-hard.[?]

4.2 The Graph Vertex Problem

- all bundles have size 2, $|S_i| = 2$

For bipartite graphs, we can get a 2-approximation and for general graphs, we can reduce them to the bipartite case(ignoring half of the edges) and therefore obtain a 4-approximation.(F. Gavril and M. Yannakakis) It seems that 4 is tight for combinatorial approaches.(APPROX 2010) It is also 2-hard assuming Unique Games Conjecture[?] and at least $\frac{17}{16}$ otherwise(under $P \neq NP$).

4.3 Conclusion

In summary, there are a lot of interesting problems for profit maximization, especially considered in Computer Science, due to online auctions.

5 Adwords Auctions

This section of the class has been delivered as a talk by Koyel Mukherjee.

5.1 Introduction

These problems investigate how search engine companies(Google, Yahoo, MSN) decide what ads to display with each query such that they maximize their revenue.(a profit maximization problem) The user types in a certain keyword(adword) in which different advertisers are interested. When a keyword comes in, the ad slots to be shown up in the search results are instantly sold to interested advertisers, by conducting an auction. Since the keywords are not known in advance, and as and when a keyword comes in, the bids by different advertisers get revealed, the nature of the auction is *online*.

We assume a completely adversarial setting, by which we mean that we make no assumption on the input pattern of keywords. Also, ad slots need to be sold instantly as and when a keyword arrives, and the decision is irrevocable. We also assume in this talk, that there is only one ad slot that needs to be auctioned, and also, the advertisers bid their true value. That is, ensuring truthfulness is not our objective in this talk. However, the advertisers have a fixed budget, and we cannot exceed the budget in our allocations.

In this talk we outline the procedure given by Buchbinder, Jain and Naor [?]. They give a primal dual mechanism achieving a competitive ratio of $1 - \frac{1}{e}$ asymptotically, matching the ratio given earlier by Mehta et al. [?]. However, the analysis presented by Buchbinder et al. is easier to understand, and does not use a tailor-made potential function for analysis, as used by Mehta et al. One crucial assumption in both these works is that the individual bid is small compared to the budget of any advertiser, in other words, the budgets are very large.

5.2 Setting

The underlying setting can be thought of as a *bipartite graph*.

- One side of the bipartition consists of the nodes corresponding to the advertisers. These sets of vertices are known to the system from the start. Let this set of advertisers be I, and $|I| = n$.
- The other side of the bipartition consists of nodes corresponding to the keywords, and these nodes along with the edges incident on them get revealed in an online manner, when they arrive. Let the set of keywords be J, and $|J| = m$.
- Each advertiser has a daily budget B_i, known to the system.

We want to assign each online arriving keyword node to one interested advertiser, who has an edge to this keyword node, and we want to maximize the weight of the matching. Once we decide on including an edge in the matching, we cannot change our decision. However, we should not exceed the budget of the advertiser. The total money that the advertiser will pay us is his budget, even if he bids in excess of his budget. If all the bids were restricted to be 0 or 1 and the budgets very large $\to \infty$, then this becomes the online b-matching problem, for which Kalyanasundaram and Pruhs [?] had proved that the best competitive ratio one can get is $1 - \frac{1}{e}$, and they gave an algorithm BALANCE that achieves this competitive ratio. Mehta et al. [?] proved that $1 - \frac{1}{e}$ is tight even for the online auctions case, and present an algorithm that asymptotically achieves this ratio, when budgets are very large. Buchbinder et al. also give an algorithm with the same competitive ratio achieved asymptotically, but with a cleaner primal-dual analysis.

The *offline primal problem* is the following:

$$\max \sum_i \sum_j b_{ij} y_{ij}$$

$$\text{s.t.}:$$

$$\sum_i y_{ij} \leq 1 \ \forall j \in J,$$

$$\sum_j b_{ij} y_{ij} \leq B_i \ \forall i \in I,$$

$$y_{ij} \geq 0 \ \forall i \in I, j \in J$$

This was proved to be NP-hard.
The *dual* for the above problem is:

$$\min \sum_i B_i x_i + \sum_j z_j$$

$$\text{s.t.}:$$

$$b_{ij} x_i + z_j \geq b_{ij},$$

$$x_i, z_j \geq 0 \ \forall i \in I, \ \forall j \in J$$

5.3 Idea

We want to use *weak duality* to bound the competitive ratio. Since the primal problem is maximization, by weak duality, any feasible primal solution P and any feasible dual solution D is related as $\mathsf{P} \leq \mathsf{D}$. This is true even for the primal optimum solution, OPT. Therefore $\mathsf{D} \geq \mathsf{OPT}$.

- We want start from 0 value both the primal and dual solutions, and at every round of online auction(keyword arrival), we increment our primal solution by $\Delta\mathsf{P}$ and dual solution by $\Delta\mathsf{D}$.
- At the end our primal is $\sum \Delta\mathsf{P}$ and the dual is $\sum \Delta\mathsf{D}$.
- We will try to bound the ratio of $\Delta\mathsf{P}/\Delta\mathsf{D}$ at every round.
- Since the ratio of $1 - \frac{1}{e}$ is tight for this problem, if we can bound

  $$\Delta\mathsf{P}/\Delta\mathsf{D} \geq 1 - \frac{1}{e}$$

  at every round, then at the end our primal solution is

  $$\mathsf{P} \geq (1 - \frac{1}{e})\mathsf{D} \geq (1 - \frac{1}{e})\mathsf{OPT}$$

  and we get an algorithm with a competitive ratio $1 - \frac{1}{e}$.

One might wonder why we can't use greedy in this problem and just allocate the keyword to the highest bidder, but due to the budget constraint we can construct an example where greedy can only do as well as $1/2$. Say, there are two types of keyword, w_1 and w_2, and two bidders each with a budget of N. Bidder 1 has bids 1 for both w_1 and w_2, whereas bidder 2 has bid 1 only for w_1 and 0 for w_2. If now w_1 comes N times, followed by w_2 N times, we allocate greedily w_1 to bidder 1 for all the N arrivals, and exhaust his budget. When w_2 comes N times, we have no one to allocate. So we get a revenue of N, whereas OPT could have got 2N.

Hence the intuition is to balance the allocation somehow. We should not allocate to the same bidder again and again. Mehta et al. [?] had a similar idea, and used a potential function to decrease the effective bid of an advertiser, depending on the how much his budget is already exhausted. Here, we outline the approach of Buchbinder et al. [?]. In this approach we assume only one keyword arrives at a time.

5.4 Algorithm

The algorithm is as follows:

1. Initially $\forall i, x_i \leftarrow 0$. (Implicitly all z_j and all y_{ij} at 0 to maintain the initial $\mathsf{P} = 0$ and $\mathsf{D} = 0$.

2. Upon arrival of a new keyword j, allocate to the advertiser $i \in \text{argmax}_{\hat{i} \in I} b_{\hat{i}j}(1 - x_{\hat{i}})$
3. If $x_i \geq 1$, do nothing.
4. Otherwise, charge i the minimum of b_{ij} and his budget, and set $y_{ij} \leftarrow 1$.
5. $z_j \leftarrow b_{ij}(1 - x_i)$ (explicitly modifying z_j only the one time it arrives. Each arrival is considered new)
6. $x_i \leftarrow x_i(1 + \frac{b_{ij}}{B_i}) + \frac{b_{ij}}{(c-1)B_i}$, where $c = (1 + R_{max})^{\frac{1}{R_{max}}}$, where $R_{max} = \max_{i \in I, j \in J} \frac{b_{ij}}{B_i}$

5.5 Analysis

Theorem 2 *The algorithm is* $(1 - \frac{1}{c})(1 - R_{max})$ *competitive for the online budgeted allocation problem. The competitive ratio* $\to \infty$ *as* $R_{max} \to 0$*, in other words, as the budgets* $B_i \to \infty$ $\forall i \in I$ *and the bids are small compared to budgets.*

Proof:

Claim 1: The algorithm produces a dual feasible solution.

We have that $x_i \geq 0$ by assignment, since we only increment x_i, if we change it. Also, $z_j \geq 0$ by assignment. We update the value of z_j to $b_{ij}(1 - x_i)$, only if $x_i < 1$, and hence $(1 - x_i) > 0$. If $x_i \geq 1$, then the dual constraint $b_{ij}x_i + z_j \geq b_{ij}$ is automatically satisfied. For the advertisers for which $x_i < 1$, we update z_j to $\max_{\hat{i} \in I} b_{\hat{i}j}(1 - x_{\hat{i}})$.

$$z_j = b_{ij}(1 - x_i) \geq b_{\hat{i}j}(1 - x_{\hat{i}}) \forall \hat{i} \in I;$$

$$z_j + b_{\hat{i}j}x_{\hat{i}} \geq b_{\hat{i}j} \forall \hat{i} \in I$$

Claim 2: $\Delta P \geq \Delta D(1 - \frac{1}{c})$ in every iteration when there is a non-zero increment in ΔD and ΔP.

When there is a non-zero increment in the dual, we have:

$$\Delta D = B_i \Delta x_i + z_j = B_i(\frac{b_{ij}x_i}{B_i} + \frac{b_{ij}}{(c-1)B_i}) = b_{ij}(1 + \frac{1}{c-1}).$$

The increment in primal is b_{ij}, because we set $y_{ij} = 1$ even if the remaining budget is less than the current bid. Hence $\Delta P / \Delta D = 1 - \frac{1}{c}$.

Claim 3: The algorithm produces an almost feasible primal solution.

We have that $y_{ij} \geq 0$ always. However, the infeasibility may arise due to violation of budget, when the remaining budget is less than the bid, and we still set $y_{ij} = 1$. In such cases, $\sum_j b_{ij} y_{ij} \geq B_i$. Ideally we want x_i to become 1 when the budget is just exhausted, but because of arbitrary values of bids, it is difficult to ensure that. Instead, we prove a weaker claim: When $\sum_j b_{ij} y_{ij} \geq B_i$, then $x_i \geq 1$. This will ensure, that the budget may be violated in at most one iteration for every bidder. We show this by proving inductively:

$$x_i \geq \frac{c^{\frac{\sum_j b_{ij} y_{ij}}{B_i}} - 1}{c-1}.$$

Hence, when the budget gets exhausted, $x_i \geq 1$. Initially it is true trivially. Let us assume it holds for bidder i at some interation k, when bidder i is chosen.

$$x_i(end) = x_i(start)(1 + \frac{b_{ik}}{B_i}) + \frac{b_{ik}}{(c-1)B_i};$$

$$x_i(end) \geq \frac{c^{\frac{\sum_{j \in J - \{k\}} b_{ij} y_{ij}}{B_i}} - 1}{c-1}(1 + \frac{b_{ik}}{B_i}) + \frac{b_{ik}}{(c-1)B_i}$$

$$x_i(end) \geq \frac{c^{\frac{\sum_{j \in J - \{k\}} b_{ij} y_{ij}}{B_i}}(1 + \frac{b_{ik}}{B_i}) - 1}{c-1}$$

$$x_i(end) \geq \frac{c^{\frac{\sum_{j \in J - \{k\}} b_{ij} y_{ij}}{B_i}} c^{\frac{b_{ik}}{B_i}} - 1}{c-1}$$

$$x_i(end) \geq \frac{c^{\frac{\sum_j b_{ij} y_{ij}}{B_i}} - 1}{c-1}$$

where the first inequality follows from induction hypothesis, and the second one from the fact: for $0 < x \leq y \leq 1$, $\frac{\ln(1+x)}{x} \geq \frac{\ln(1+y)}{y}$. (easy to prove). In this inequality, if we replace x with $\frac{b_{ik}}{B_i}$ and y by R_{max}, then we get,

$$\frac{\ln(1 + \frac{b_{ik}}{B_i})}{\frac{b_{ik}}{B_i}} \geq \frac{\ln(1 + R_{max})}{R_{max}}$$

But $\ln(1 + R_{max}) = \ln(c^{R_{max}}) = R_{max} \ln(c)$. It is easy to get the rest.

However, we still might have violated the budget in at most one iteration for every $i \in I$. The maximum violation is: $\sum_j b_{ij} y_{ij} \leq B_i + \max_{j \in J} b_i j$. Therefore, we can lower bound the real (feasible) primal profit obtained by the algorithm as $\sum_j b_{ij} y_{ij} \frac{B_i}{B_i + \max_{j \in J} b_{ij}} \geq \sum_j b_{ij} y_{ij} \frac{1}{1+R_{max}} \geq \sum_j b_{ij} y_{ij}(1 - R_{max})$. The first

inequality is by the definition of R_{max} and the second inequality is by binomial expansion.
Putting the three claims together, we have that our algorithm gives a feasible primal value $P_f \geq P(1 - R_{max}) \geq D(1 - \frac{1}{c})(1 - R_{max}) \geq (1 - \frac{1}{c})(1 - R_{max})OPT$, at the end, thereby proving the theorem. ■

References

[1] N. Buchbinder, K. Jain and J. Naor, *Online primal-dual algorithms for maximizing ad-auctions revenue*, **Algorithms- ESA**, 253-264,2007

[2] E.D. Demaine, U. Feige, M.T. Hajiaghayi and M.R. Salavatipour,*Combination can be hard: approximability of the unique coverage problem*,**SIAM J.Comput.**,38(4):1464-1483,2008

[3] K.M. Elbassioni, R. Raman and S. Ray, *On profit-maximizing pricing for the highway and toolbooth problems*,**CoRR**, abs/0901.1140, 2009

[4] I. Gamzu and D. Segev, *A subalgorithmic approximation for highway and tollbooth pricing*,textbfCoRR, abs/1002.2084, 2010

[5] F. Grandoni and T. Rhotvoss, *Prizing on Paths:A PTAS for the Highway Problem*,**CoRR**, abs/1004.3051, 2010

[6] Hajiaghayi, Mohammad T. *Algorithmic Game Theory: Lecture 5– Handwritten notes.* 29 September 2010. University of Maryland.

[7] B. Kalyanasundaram and K. Pruhs, *An optimal deterministic algorithm for online b-matching*, **Foundations of Software Technology and Theoretical Computer Science**, 193-199,1996

[8] S. Khot and O. Regev, emphVertex cover might be hard to approximate to within $2 - \epsilon$,**Journal of Computer and System Sciences**, 74(3): 335-349, 2008

[9] A. Mehta, A. Saberi, U. Vazirani and V. Vazirani, emphAdwords and generalized online matching, **Journal of the ACM**, 54(5):22, 2007