Allocating online advertisement space with unreliable estimates

Mohammad Mahdian
Yahoo! Research
mahdian@yahoo-inc.com

Hamid Nazerzadeh
Stanford University
hamidnz@stanford.edu

Amin Saberi
Stanford University
saberi@stanford.edu

Abstract

We study the problem of optimally allocating online advertisement space to budget-constrained advertisers. This problem was defined and studied from the perspective of worst-case online competitive analysis by Mehta et al. Our objective is to find an algorithm that takes advantage of the given estimates of the frequencies of keywords to compute a near optimal solution when the estimates are accurate, while *at the same time* maintaining a good worst-case competitive ratio in case the estimates are totally incorrect. This is motivated by real-world situations where search engines have stochastic information that provide reasonably accurate estimates of the frequency of search queries except in certain highly unpredictable yet economically valuable spikes in the search pattern.

Our approach is a black-box approach: we assume we have access to an oracle that uses the given estimates to recommend an advertiser every time a query arrives. We use this oracle to design an algorithm that provides two performance guarantees: the performance guarantee in the case that the oracle gives an accurate estimate, and its worst-case performance guarantee. Our algorithm can be fine tuned by adjusting a parameter α, giving a tradeoff curve between the two performance measures with the best competitive ratio for the worst-case scenario at one end of the curve and the optimal solution for the scenario where estimates are accurate at the other end. Finally, we demonstrate the applicability of our framework by applying it to two classical online problems, namely the lost cow and the ski rental problems.

1 Introduction

Search engines such as Google, Yahoo!, and MSN use a simple but innovative auction mechanism for allocating the advertisement space on the side of search results. Goods that are being sold in these auctions, the search queries, have very interesting characteristics; most importantly, they should be allocated to the buyers in a fraction of a second or they will perish instantly. Moreover, they arrive in an online fashion and their total supply, the number of times the users search for a specific keyword, is unknown. The online nature of these auctions gives rise to interesting and challenging algorithmic problems. One of the most central problems in this context is finding the optimum allocation algorithm: given the current bids and budgets of the advertisers, what is the best algorithm for allocating each search query to the advertisers?

Mehta et al. [17] first posed this problem as a generalization of the online matching problem: the search engine is given the bids and the budgets of advertisers. As a search query arrives, the ad space is allocated to the advertisers and they are charged their bid for an impression. The goal is to compute the allocation in such a way that maximizes the revenue while respecting the budget constraints of the advertisers. Mehta et al. [17] gave an algorithm with the competitive ratio of $(1 - 1/e)$ and showed that this is the best possible ratio an online algorithm can achieve. They also presented extensions of their algorithm to more realistic settings used in practice.

It has been argued by many (see, for example, Koutsoupias and Papadimitriou [14]) that the pessimistic nature of the worst-case analysis for online algorithms favors algorithms that make extremely conservative choices. In practice, as they point out, we usually have some information about the input. For example, in our context it is reasonable to assume that the search engines have a good estimate of the frequencies of keywords. The main problem is that although these estimates are often accurate, they might turn out to be completely wrong due to unexpected events. Examples include the sudden spike in the frequency of search for "gas mask" right after speculations about terrorist activities, or for "Janet Jackson" after the wardrobe malfunction incident in Super Bowl 2004, or simply the surge in the volume of associated queries after the release of a new successful movie or record. Such spikes are usually very valuable for advertisers (and hence for the search engine), and at the same time they are essentially impossible to predict. As we will observe in the next section, an allocation algorithm that decides entirely based on the estimates might completely miss out on such revenue opportunities.

To address this problem, we define a new framework for designing and evaluating online algorithms in presence of unreliable estimates. Our goal is to design an algorithm that performs well in the case that the estimates are accurate as well as in the worst case. We evaluate our algorithm using two parameters: the worst-case competitive ratio γ_w, and the competitive ratio γ_a in the case the given estimates are accurate. In fact, for the query allocation problem, we prove a *stronger* result: we show that given *any* oracle that makes a recommendation to the algorithm in every step, we can design an algorithm that allocates queries in such a way that the value of the solution is always at least a fraction γ_w of the optimal offline solution, *and* a fraction γ_a of the solution provided by the oracle. For example, if estimates of frequencies of keywords are available, the oracle can be designed to compute the optimal solution given these frequencies by solving a linear program, in which case our algorithm provides a performance guarantee of γ_a with respect to the optimal solutions on inputs conforming with the predicted frequencies.

We propose a family of algorithms parameterized by a value $\alpha \geq 1$. This parameter, roughly speaking, reflects our trust in the estimates. If $\alpha = 1$, our algorithm ignores the oracle and will have the same performance of $1 - 1/e$. If $\alpha = \infty$, the algorithm will always follow the recommendations of the oracle. We give a tight analysis of our algorithm for all values of α. Our analysis is based on using a factor-revealing program [11, 12]. For deriving γ_w, the structure of this program is similar to the one used by Mehta et al. [17], and can be solved in a similar fashion. However, the structure of the factor revealing LP for computing γ_a becomes much more complex and more challenging to analyze. Even though in a certain range, the competitive ratio γ_a can only be described as the solution of an algebraic equation, we can still produce examples to show that our analysis is tight.

We believe that the framework introduced in this paper can be useful in other online computing problems, when the decision maker has unreliable information about the future. In the last section, we show how this framework can be applied to two textbook examples of online algorithms, namely the lost cow problem and the ski rental problem.

Related work. Mehta et al. [17] build their results upon the work of Karp, Vazirani, and Vazirani [13] on the online matching problem, and the work of Kalyanasundaram and Pruhs [10] on the online b-matching problem. Recently, Buchbinder, Jain, and Naor [8] gave a simple and intuitive primal-dual interpretation of the algorithm of Mehta et al. [17], and provide several generalizations, including one that claims to take advantage of stochastic information. However, their model and the type of the stochastic information they use is entirely different from ours. Both our results and the results in [17, 8] deal with the algorithmic, rather than the mechanism design aspect of the allocation problem. Designing a revenue-competitive incentive compatible mechanism for this problem seems to be a difficult task, and is only solved for the case of multi-unit auctions (i.e., when each advertiser bids on only one keyword) by Mahdian and Saberi [16].

Ad auctions have inspired several interesting algorithmic and game theoretic questions [5, 1, **?**]. For a study of the equilibria of these games in a static setting without budgets see [9]. See [15] for a survey of results in this area.

2 The query allocation problem

In this section, we define the query allocation problem (first posed by Mehta et al. [17]), and our framework for incorporating unreliable estimates. In this model, the search engine receives the bids of advertisers for each keyword and their total budget for a certain period (e.g. a day). During the day, as a search query for a keyword arrives, the search engine assigns one advertiser to this query, and charges this advertiser an amount equal to their bid for the corresponding keyword. Note that, as in [17], we make the simplifying assumption that there is only one ad slot on each search results page. As Mehta et al. [17] point out, their result (and similarly ours) can be easily extended to more realistic settings where there are several slots on each page, and the search engine charges the advertisers per click instead of per impression.

Let $\mathcal{A}$ be the set of advertisers, $\mathcal{K}$ be the set of keywords, and $\mathcal{Q}$ be the sequence of queries, which is given to the algorithm in an online fashion. Each query in $\mathcal{Q}$ is for a keyword in $\mathcal{K}$. Denote the budget of advertiser i by B_i, and the bid of advertiser i for keyword j by b_{ij}. As in [17], we assume that bids are small compared to the budgets. The goal is to maximize the revenue, i.e., the sum of $b_{i(q),j(q)}$ over all queries q in $\mathcal{Q}$, where $i(q)$ denotes the advertiser assigned to this query, and $j(q)$ denotes the keyword corresponding to this query. Therefore, if we know the number of times n_j a query for keyword j occurs in $\mathcal{Q}$, the solution of this problem can be computed by solving the following maximization program with integrality constraints on x_{ij}'s:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathcal{A}, j \in \mathcal{K}} b_{ij} x_{ij} \\ \text{subject to} \quad & \forall j \in \mathcal{K}: \ \sum_{i \in \mathcal{A}} x_{ij} \le n_j \\ & \forall i \in \mathcal{A}: \ \sum_{j \in \mathcal{K}} b_{ij} x_{ij} \le B_i \\ & \forall i \in \mathcal{A}, \forall j \in \mathcal{K}: \ x_{ij} \ge 0 \end{aligned} \tag{1}$$

Note that the assumption that bids are small compared to budgets implies that relaxing the integrality constraints in the above program does not change the solution of the program by a significant amount.

When frequencies n_j are unknown, Mehta et al. [17] give a $(1 - 1/e)$-competitive algorithm for this problem, and show that this is the best ratio an online algorithm can achieve.

Unreliable estimates. We assume that the search engine has access to an oracle EST which recommends an advertiser for each query. We treat this oracle as a black box; it can be based on an algorithm that solves the linear program 1 for given estimates n_j of the frequencies (if such estimates are available), or even a more complex algorithm that learns the distribution of the queries and the corresponding optimal allocation over time.

As the following example shows, the naive approach of simply following the recommendations of EST can produce very small revenue compared to the the optimal offline allocation, if EST is based on wrong estimates.

Example A naive approach is based on taking the given estimates of the frequencies n_j, computing the optimal solution S with respect to these frequencies, and allocating according to this optimal solution (or, if a query arrives whose assignment is not predicted in S, assigning this query to the highest bidder with enough remaining budget). Consider the following scenario: there are two advertisers, each with a budget of 100, and two keywords, each with the predicted frequency of 100 during a day. Advertiser 1 only bids 1 on keyword 1, and nothing on keyword 2. The bid of advertiser 2 on keywords 1 and 2 are 10 and 1, respectively. The optimal solution with our estimates of the frequencies is to assign all queries for keyword 1 to advertiser 1, and all queries for keyword 2 to advertiser 2. However, if instead, we get only 10 queries for keyword 1, following the naive approach would give a revenue of 10, while the optimal revenue is 100 (assigning all queries to advertiser 2).

The above example shows that the worst-case competitive ratio of the naive approach is zero. In fact, it is easy to see from the above example that there is no algorithm that achieves the optimal solution when the estimates are accurate *and* has a zero worst-case competitive ratio. The natural question is, if we allow a small decrease in revenue in cases where our estimates are accurate, would we be able to achieve a bounded worst-case competitive ratio?

To make this precise, we define two parameters γ_w and γ_a for evaluating an algorithm. γ_a represents the competitive ratio with respect to the solution we get if we follow the EST recommendation every time[1]. γ_w is the competitive ratio with respect to the optimal offline solution.[2]

By this definition and the above example, an algorithm that always follows EST has $\gamma_a = 1$ and $\gamma_w = 0$. On the other hand, an algorithm that always follows the online algorithm of Mehta et al. [17] has $\gamma_a = \gamma_w = 1 - 1/e$. The goal is to design an algorithm that intelligently decides when to follow EST and when not, in order to achieve a tradeoff between these two extreme points. A simplistic way to do this in expectation is by using randomization:

Example A simple way for achieving non-zero γ_w and γ_a in expectation is the following: Toss a coin and with probability p follow EST on every query. With probability $(1-p)$ ignore EST and follow the $(1-1/e)$-competitive algorithm of Mehta et al. [17]. Clearly, in expectation, this algorithm achieves a p fraction of the revenue of EST and a $(1-p)(1-1/e)$ fraction of the optimal offline revenue.

In the rest of this paper, we propose and analyze an algorithm that performs better than the above approach (and in addition lacks the undesirable feature that the output is a random variable with high variance – in fact, our algorithm will be deterministic).

3 The Algorithm

The main idea of the algorithm is to rely on EST's recommendation only when it is, roughly speaking, "safe". To explain the intuition behind the algorithm, we need to quickly present the $(1-1/e)$-competitive algorithm of Mehta et al [17]. Their algorithm uses a function $\Phi(f) = 1 - e^{f-1}$ that takes as the input the fraction of the budget of an advertiser that is spent, and outputs a factor by which the bid of the advertiser should be discounted. As a new query arrives, the algorithm assigns it to the bidder with the maximum discounted bid.

Our algorithm is parameterized by a number $\alpha \geq 1$. This parameter controls the extent to which we want to rely on the EST. For $f \in [0,1]$, define $\Phi(f) := 1 - e^{\alpha\cdot(f-1)}$. Also, let f_i be the fraction of the budget of advertiser i that has been spent so far. As a new query for keyword j arrives, the algorithm finds the advertiser d that maximizes $\Phi(f_d)b_{dj}$ over all $d \in \mathcal{A}$. Also, suppose EST's recommendation is to allocate the query to the advertiser e. The algorithm compares $\Phi(f_e)b_{ej}$ with $\Phi(f_d)b_{dj}$. If the latter is bigger than the former by more than a factor of α, then the query is allocated to e; otherwise, it is allocated to d.

This algorithm is shown in Figure 1. In the next two sections, we analyze the performance guarantees γ_w and γ_a of the algorithm. We start with the analysis of γ_w, which is simpler and follows the same line as in [17].

4 Competitive ratio when the estimates are inaccurate

In this section, we find γ_w, the competitive ratio of our algorithm with respect to the optimal offline solution, independent of the accuracy of the estimates. The analysis in this section is very similar to [17] and therefore we will discuss it only briefly. Let us start by some definitions. Let k be a large number we use for discretizing the budgets of the advertisers. Let

$$\phi(s) = 1 - (1 - \frac{1}{k})^{\alpha(k-s)}. \tag{2}$$

Note that when k tends to infinity we can approximate the function $\phi(s)$ with $\Phi(\frac{s}{k})$. We say that advertiser i is of type t, $0 \leq t \leq k$, if she has spent within $(\frac{t-1}{k}, \frac{t}{k}]$ fraction of her budget so far. Let s_t be the total budget

[1] Notice that we do not assume that the solution of EST is necessarily optimum.

[2] Note that by this definition, we always have $\gamma_w \geq \gamma_a$.

Algorithm:

Upon the arrival of a new query for keyword j:

- Let e be the advertiser recommended by EST for receiving j;
- Let d be the advertiser with maximum $\Phi(f_i)b_{ij}$ among all $i \in \mathcal{A}$;
- If $\alpha\Phi(f_e)b_{ej} \geq \Phi(f_d)b_{dj}$ then

  allocate j to e;

  else

  allocate j to d;

Figure 1: The algorithm

of these advertisers. We also define r_t to be the total remaining budget of the advertisers of type t in the optimal solution, OPT. Also, let S be total budget of the advertisers. For $i = 0, 1, \ldots, k$, define w_i to be the amount of money spent by all the advertisers from the interval $(\frac{i-1}{k}, \frac{i}{k}]$ fraction of their budgets. Using these definitions we state lemmas below. The first lemma can be proved similarly to lemma 7 in [17]. Lemma 2 is proved by simply summing up a geometric series. The proofs of these lemmas are omitted here.

Lemma 1 *At the end of the algorithm, we have:*

$$\sum_{i=0}^{k} \phi(i)(s_i - r_i) \leq \sum_{i=0}^{k} \alpha\phi(i)w_i \tag{3}$$

Lemma 2

$$\frac{1}{k}\sum_{t=j}^{i} \phi(t) = \frac{i-j}{k} + \frac{1}{\alpha}(\phi(i) - \phi(j)) + O(1/k) \tag{4}$$

Now, we are ready to prove the main result of this section.

Theorem 3 *The competitive ratio of the algorithm with respect to optimum offline solution, γ_w, is $\frac{1}{\alpha}(1 - \frac{1}{e^\alpha})$.*

Proof : By definition w_i is at most $\frac{1}{k}\sum_{j=i}^{k} s_j$. Plugging in (3):

$$\sum_{i=0}^{k} \phi(i)(s_i - r_i) \leq \alpha \sum_{j=0}^{k} \phi(j)(\frac{1}{k}\sum_{i=j}^{k} s_i) \leq \frac{\alpha}{k}\sum_{i=0}^{k} s_i \sum_{j=0}^{i} \phi(i) \tag{5}$$

Using Lemma 2 we get:

$$(\phi(0) - O(1/k))S - \sum_{i=0}^{k} \phi(i)r_i \leq \alpha \sum_{i=0}^{k} \frac{i}{k} s_i \tag{6}$$

Note that $\sum_{i=0}^{k} \frac{i}{k} s_i$ is the revenue of our algorithm and $S - \sum_{i=0}^{k} r_i$ is the revenue of OPT, which is less than or equal to $\phi(0)S - \sum_{i=0}^{k} \phi(i)r_i$.

Tight Example: We construct a scenario in which the revenue of the optimal solution is 1 and ours is γ_w. Suppose that there are k advertisers with budget $\frac{1}{k}$. The keywords show up in k phases. In the phase i, the keywords of advertiser i arrive for which she bids $\alpha\epsilon$ and in the optimal solution these keywords are allocated to i. But our algorithm, based on EST, distributes these keywords among advertisers $i+1 \dots k$ with bid ϵ. It is easy to see that in this scenario the revenue of the algorithm matches the lower bound in the theorem. □

Example: For α equal to $1.5, 1.75$ and 2, γ_w is equal to 0.52, 0.47 and 0.43.

5 Competitive ratio when the estimates are accurate

In this section, we evaluate γ_a, the competitive ratio of the algorithm with respect to the allocation defined by the recommendation of EST. Similar to the previous section, we define r_i to be the total remaining budget of advertisers of type i in the solution of EST. Also, we assume that the budgets and the bids are normalized so that $S = 1$.

We find an upper bound on the value of the keywords we may miss by ignoring the EST's recommendations. Let $\mathcal{C}$ be the set of all keywords for which we made a different choice than EST. We define t_{ij} to be the total amount of budget that advertisers of type i have spent from interval $(\frac{j-1}{k}, \frac{j}{k}]$ fraction of their budget for the keywords in $\mathcal{Q}-\mathcal{C}$, the set of keywords for which both algorithms made the same decision. Recall from the algorithm that when the keyword j shows up, j is allocated to EST's candidate, e, if $\alpha\Phi(f_e)b_{ej} \geq \Phi(f_d)b_{dj} = \max_{i\in\mathcal{A}} \Phi(f_i)b_{ij}$. An important observation here is for every keyword $j \in \mathcal{C}$:

$$\Phi(f_e)b_{ej} < \frac{1}{\alpha}\Phi(f_d)b_{dj}$$

Now, summing up over all queries in $\mathcal{C}$ we get:

$$\sum_{i=0}^{k} \phi(i)(s_i - r_i - \sum_{j=0}^{i} t_{ij}) \leq \frac{1}{\alpha}\sum_{i=0}^{k} \phi(i)(w_i - \sum_{j=i}^{k} t_{ji})$$

Plugging $w_i = \frac{1}{k}\sum_{j=i}^{k} s_j$ and rearranging the terms we get the inequality below.

$$\sum_{i=0}^{k} \phi(i)(\frac{1}{\alpha}\sum_{j=i}^{k} t_{ji} - \sum_{j=1}^{i} t_{ij}) \leq \frac{1}{\alpha}\sum_{i=0}^{k} \phi(i)(\frac{1}{k}\sum_{j=i}^{k} s_j) - \sum_{i=0}^{k} \phi(i)(s_i - r_i) \tag{7}$$

The inequality (7) is derived in the same way as inequality (5) except that α has been replaced by $\frac{1}{\alpha}$ in the right hand side. The big difference however, is made in the left hand side of in equality (7). The variables t_{ij} may make the left hand side of (7) negative or positive, which dramatically changes the structure of the worst-case examples.

To find the competitive ratio of the revenue of our algorithm with respect to EST, we use the factor revealing LP technique [11, 12]. Consider the LP below. The objective function is the revenue of the algorithm and the linear constraints are imposed by the algorithm and the structure of the problem. Therefore, any feasible solution of this LP gives an upper bound on the competitive ratio of the algorithm.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=0}^{k} \frac{i}{k} s_i && (8)\\
\text{subject to} \quad & \sum_{i=0}^{k} s_i = S \\
& \sum_{i=0}^{k} \phi(i)(\frac{1}{\alpha}\sum_{j=i}^{k} t_{ji} - \sum_{j=1}^{i} t_{ij}) \quad \leq \quad \sum_{i=0}^{k} \phi(i)(\frac{1}{\alpha k}(\sum_{j=i}^{k} s_j) - s_i + r_i) && (9)\\
& \forall 0 \leq j \leq i \leq k: \ \frac{1}{k} s_i \geq t_{ij} \\
& \sum_{i=0}^{k} r_i = R \\
& \forall i: \ s_i \geq r_i \\
& s_i, r_i, t_{ij} \geq 0
\end{aligned}
$$

Using Lemma 2, we get the dual of this linear program:

$$
\begin{aligned}
\text{maximize} \quad & xS - vR && (10)\\
\text{subject to} \quad & \forall 0 \leq j \leq i \leq k: \ (\alpha\phi(i) - \phi(j))y \leq z_{ij} \\
& \forall i: \ x + (\frac{i}{k} - \frac{1}{\alpha}\phi(0) + (\frac{1}{\alpha} - \alpha)\phi(i))y + \frac{1}{k}\sum_{j=0}^{i} z_{ij} + u_i \leq \frac{i}{k} && (11)\\
& \forall i: \ y\alpha\phi(i) - u_i \leq v \\
& y, z_{ij}, u_i \geq 0
\end{aligned}
$$

We formalize this in the theorem below:

Theorem 4 *Let α^* be the root of $(\alpha^2 - 1/\alpha)e^{-\alpha} + 1/\alpha - 1 = 0$ in $[1, 2]$* [3]*. For $\alpha \geq \alpha^*$, let*

$$
\gamma_a = \frac{\alpha(e^\alpha - 1)}{(\alpha - \frac{1}{\alpha})(e^\alpha - 1) + e^\alpha}. \tag{12}
$$

Also, let f^* be the solution for the equation $(\alpha(f-1))^2 e^{\alpha(f-1)} = 1 - \frac{1}{\alpha}(1 - e^{-\alpha})$. For α in $(1, \alpha^*)$, let

$$
\gamma_a = 1 - \frac{1 - \frac{1}{\alpha}(1 - e^{-\alpha})}{\alpha(1 - (1 + \alpha(f^* - 1))e^{\alpha(f^*-1)})} \tag{13}
$$

Then on any input, our algorithm always produces a solution whose value is at least a γ_a fraction of the value of the solution found by EST.

Proof : We start by constructing a feasible dual solution for (10). Set $v = y\alpha\phi(0)$, $u_i = 0$, $z_{ij} = max\ \{(\alpha\phi(i) - \phi(j))y, 0\}$, for all $1 \leq j \leq i \leq k$. Now, the only unsatisfied condition is (11). Let $j^* = \min_{j \geq 0}\{\alpha\phi(i) - \phi(j) \geq 0\}$. Taking sum over all inequalities for z_{ij}, by Lemma 2, we have:

$$
\begin{aligned}
\frac{1}{k}\sum_{j=0}^{i} z_{ij} &\geq \frac{1}{k}\sum_{j=j^*}^{i} (\alpha\phi(i) - \phi(j))y \\
&= ((\alpha\frac{i-j^*}{k} - \frac{1}{\alpha})\phi(i) + \frac{1}{\alpha}\phi(j^*) - \frac{i-j^*}{k})y
\end{aligned}
$$

[3] α^* is very close to 1.795

Suppose k tends to infinity. Let $f = \frac{i}{k}$, recall $\Phi(f) = (1 - e^{\alpha(f-1)})$. If $\alpha\Phi(f) \geq \Phi(0)$, then $j^* = 0$, so by (11) we have:

$$x \leq f - \alpha(f-1)\Phi(f)y \tag{14}$$

If $\alpha\Phi(f) < \Phi(0)$, then $\frac{j^*}{k} = \frac{1}{\alpha}\ln(1 - \alpha(1 - e^{\alpha(f-1)})) + 1$. Plugging into (11) we get:

$$\begin{aligned} x &\leq \frac{i}{k} - (\alpha(\frac{i-j^*}{k} - 1)\phi(i) + \frac{1}{\alpha}\phi(j^*) - \frac{1}{\alpha}\phi(0) + \frac{j^*}{k})y \\ &= f - ((\alpha(f-1)+1)\Phi(f) - \frac{1}{\alpha}\Phi(0) + (\frac{1}{\alpha}\ln(1-\alpha\Phi(f)) + 1)(1-\alpha\Phi(f)))y \end{aligned} \tag{15}$$

We find the values of f for which r.h.s in (14) and (15) take their minimums. Taking the first and second derivatives of the $f - \alpha(f-1)\Phi(f)y$, the function is concave for $f \leq 1 - \frac{2}{\alpha}$ and it is strictly convex after that. Also, the r.h.s in (15) is a concave function of f. For $\alpha^* \leq \alpha$ this function is increasing at 0. So for $\alpha \geq \alpha^*$, we can choose the value of y such that the r.h.s takes its minimum at $f = 0$ and $f = 1$. Hence, the corresponding inequalities hold as equalities. This leads to following feasible solution:

$$x = \alpha\Phi(0)y = 1 - (1 - \frac{1}{\alpha}\Phi(0))y \Rightarrow \begin{cases} y = \frac{1}{(\alpha - \frac{1}{\alpha})\Phi(0)+1} \\ \\ x = \frac{\alpha\Phi(0)}{(\alpha-\frac{1}{\alpha})\Phi(0)+1} \end{cases}$$

The case for $1 < \alpha < \alpha^*$ is different because the function $f - \alpha(f-1)\Phi(f)y$ in r.h.s of inequality (14), is strictly convex in for $0 \leq f \leq 1$, and it is decreasing at point 0. So the function takes it minimum at the point $f^* \neq 0$ which satisfies the first order conditions:

$$1 - \alpha(1 - (1 + \alpha(f^* - 1))e^{\alpha(f^*-1)})y = 0$$

So we choose the value of y that makes the minimum in r.h.s. of (14) (for f^*) and r.h.s (15) (for 1) equal. The solution of this equation gives the value of x for $1 < \alpha < \alpha^*$.

Tight Example: Now we give a tight example for this analysis where $\alpha > \alpha^*$. Suppose there are two advertisers with budgets $B_1 = \frac{1}{\alpha\Phi(0)}(1 - \frac{1}{\alpha}\Phi(0))$ and $B_2 = 1$. First, the keywords of advertiser 1 arrive and in the solution proposed by EST they are allocated to 1. Suppose when keyword j in this set shows up, $b_{1j} = \epsilon$ and b_{2j} is slightly more than $\frac{\alpha}{\Phi(f_2)}\epsilon$. So the algorithm allocates j to 2. This continues up to the time advertiser 2 runs out of budget. After that, a set of keywords show up that only 2 is interested but she has no budget to take them. In this case $\gamma_a = \frac{B_2}{B_1+B_2}$ which matches the lower bound. Also, this corresponds to a feasible primal solution with $s_0 = B_1$ and $s_k = B_2$ and $t_{ij} = 0$ for all $1 \leq j \leq i \leq k$. For the case $\alpha < \alpha^*$, the optimal primal solution has a similar structure except that $S_{f^*} = B_1$ and $t_{ij} = \frac{1}{k}B_1$ for $i = \lfloor kf^* \rfloor, j \leq i$. $\square$

Example: For α equal to $1.5, 2$ and 5, γ_a is $0.71, 0.75$ and 0.86.

6 Extensions of the Framework

A common criticism of the framework of online competitive analysis is that it is too pessimistic: it assumes that we have no knowledge of the sequence of events in the future, whereas in practice we often have a good estimate of events that might happen in the future. The notion we introduced in this paper is an attempt to answer this problem. The idea is to give algorithms with a considerably better competitive ratio if the future events are actually close to our estimates, while still maintaining a reasonable competitive ratio, in case an unpredicted event changes the sequence of events drastically. This concept can be applied to many online problems. One prominent example was the advertisement allocation problem that was presented in the previous sections. In this section, we present two other examples - the well-known lost-cow and ski-rental problems - to further illustrate the concept.

Ski-Rental problem: the parents of a teenager are taking her skiing. Every time they go skiing, they are faced with the decision of buying or renting the ski equipments. Buying ski equipments costs a times as much as renting them. The problem is that the parents do not know how many times their child will come for skiing before she loses interest. There is a simple 2-competitive algorithm to solve this problem that is usually given as the classical classroom example for online algorithms: rent ski equipments for a day, and then buy them. It is not hard to show that this is the best deterministic algorithm for this problem.

Now, assume that the parents have an estimate, b, of the number of times their child will go skiing. They would like to follow an algorithm that is close to optimal if their estimate turns out to be accurate, while still not costing them too much if the estimate is wrong. The following theorem answers this question.

Theorem 5 *For any value $1 < \alpha \leq 2$, there is an algorithm for the ski-rental problem that achieves an approximation ratio of α if the estimate b is correct, and a competitive ratio of $\alpha/(\alpha - 1)$ if it is not correct. Moreover, this is the best competitive ratio a deterministic algorithm can achieve.*

The lost cow problem A smart shortsighted cow, is standing in front of a fence and does not know where the gate is located. At each time, the cow can goes one step to the right or to the left and see if it is in front of the gate. The cow wants to find the gates in the minimum number of steps.

Suppose the cow has a guess about the place of the gate but it is not sure. Based on this estimation, we can design a family of algorithm. This idea is formalized in the proposition bellow, the proof is appeared in appendix B

Proposition 6 *For every $\alpha \geq 2$, there is an algorithm for the lost-cow problem that has competitive ratio of $2\frac{\alpha^2}{\alpha-1} + 1$ in the worst-case. If the estimate is correct, the competitive ratio is $1 + \frac{2}{\alpha-1}$.*

7 Conclusion

In this paper, we have presented a family of online algorithms for allocating online advertisement with unreliable estimates. We have used trade off revealing LP technique for designing and analyzing these algorithm in both cases for the wrong and accurate estimates. We have also provided examples which show that our analysis is tight.

The tight examples for the worst case and best case scenarios are essentially different, which suggests that choosing α randomly, in average, will improve the performance of the algorithm.

We have simulated the performance of our algorithm in a setting with 20 advertisers and about 5000 queries, with uniform distribution over the bids and the frequency of the keywords. The experimental results are significantly better than the worst-case examples we came up with in our tight analysis. As an example, for α equal to 2, with accurate estimates we almost have the same revenue as the optimal solution. Also, γ_w turned out to be more than 0.75.

Acknowledgment. We would like to thank Rajat Bhattacharjee, Nicole Immorlica, Bobby Kleinberg and an anonymous referee for very helpful comments. The second and the third authors were supported by a gift from Google.

References

[1] Z. Abrams, *Revenue Maximization When advertisers Have Budgets*, In Proc. of the ACM-SIAM Symposium on Discrete Algorithms, 2006.

[2] N. Andelman and Y. Mansour, *Auctions with budget constraints*, In Proc. of the Scandinavian Workshop on Algorithm Theory, 2004.

[3] G. Aggarwal, A. Goel, and R. Motwani, *Truthful auctions for pricing search keywords*, In Proc. of the ACM conference on Electronic Commerce, 2006.

[4] R. Baeza-Yates, J. culberson, G. Rawlins, *Searching in the plane*, J. Information Computing, 1993.

[5] C. Borgs, J. Chayes, N. Immorlica, M. Mahdian and A. Saberi, *Multi-unit auctions with budget-constrained bidders*, In Proc. of the ACM Conference on Electronic Commerce, 2005.

[6] Allan Borodin, Ran El-Yaniv *Online Computation and Competitive Analysis*, Cambridge University Press, 1998.

[7] Niv Buchbinder, Joseph Naor, *Online Primal-Dual Algorithms for Covering and Packing Problems*, In Proc. of the European Symposium on Algorithms, 2005.

[8] N. Buchbinder, K. Jain, and J. Naor, *Online Primal-Dual Algorithms for Maximizing Ad-Auction Revenue*, manuscript, 2006.

[9] B. Edelman, M. Ostrovsky, and M. Schwarz, *Internet Advertising and the Generalized Second Price Auction: Selling Billions of Dollars Worth of Keywords*, Working paper, http://rwj.berkeley.edu/schwarz/, 2005.

[10] B. Kalyanasundaram, K.Pruhs, *An optimal deterministic algorithm for online b-matching* J. Theoretical Computer Science, 2000.

[11] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. Vazirani, *Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP* . J. ACM, 2003.

[12] K. Jain, M. Mahdian, and A. Saberi, *A new greedy approach for facility location problems*, In Proc. of ACM Symposium on Theory of Computing, 2002.

[13] R. Karp, U. Vazirani, and V. Vazirani, *An optimal algorithm for online bipartite matching*, In Proc. of the ACM Symposium on Theory of Computing, 1990.

[14] E. Koutsoupias, C. H. Papadimitriou, *Beyond competitive analysis*, In Proc. of the Symposium on the Foundation of Computer Science, 1994

[15] S. Lahaie, D. Pennock, A. Saberi, and R. Vohra, *Sponsored Search*, To appear in Algorithmic Game Theory (Nisan et al. eds.), 2007.

[16] M. Mahdian, A. Saberi, *Multiunit Auctions with Unknown Supply*, In Proc. of the ACM Conference on Electronic Commerce, 2006.

[17] A. Mehta, A. Saberi, U. Vazirani, V. Vazirani, *Adwords and generalized on-line matching*, In Proc. of the Foundation of Computer Science, 2005

A Proof of Proposition 5

.

Proof : There is a simple 2-competitive algorithm to solve this problem that is usually given as the classical classroom example for online algorithms: rent ski equipments for a day, and then buy them. It is not hard to show that this is the best deterministic algorithm for this problem.

Let $\beta = \alpha/(\alpha - 1)$. Any deterministic algorithm for this problem must decide on the number of times x the parents must rent the ski equipments before they buy them. The competitive ratio of such an algorithm is given by

$$\frac{x + a}{\min(x + 1, a)}.$$

Using the above expression, it is easy to see that in order to get a competitive ratio of β, it is necessary and sufficient to pick the value x in the interval $[(a-\beta)/(\beta-1), (\beta-1)a]$. It only remains to decide which point of this interval to pick, in order to achieve a factor of at most α when the estimate b is accurate. To do so, first notice that if $b \leq a$, one can easily pick $x = a$, since in this case the 2-competitive algorithm for ski-rental is actually optimal when the estimates are accurate. Therefore, assume $b > a$. Now, notice that the algorithm that always picks $x = (a-\beta)/(\beta-1)$ has an approximation factor of at most α when the estimates are correct. This follows from the fact that the approximation factor of the algorithm when the estimates are correct can be given by

$$\begin{cases} \frac{b}{a} & \text{if } x \geq b \\ \frac{x+a}{a} & \text{if } x < b. \end{cases}$$

Finally, considering the case that b is a large number, and using the above expression and the fact that x must be selected from the interval $[(a - \beta)/(\beta - 1), (\beta - 1)a]$, one can easily see that no algorithm can achieve an approximation ratio (for the case the estimates are correct) better than $\beta/(\beta - 1) = \alpha$.

$\square$

B Proof of Proposition 6

Proof : Baeza-Yates et. al [4], present a deterministic optimal algorithm for this problem. There is a series of phases, at phase 1, the cow takes one of the left or right direction and go 1 step through that direction, if it finds the gate then it is done, other wise it comes back to its initial place and goes 2 step in other direction, and will continue doubling movement till it finds the gate.

Our solution for the problem is the following: Let d be the claimed distance to the gate, similar to [4], at each phase multiply the number of steps by α, but choose the direction in the first phase such that in phase i where $\alpha^i \leq d < \alpha^{i+1}$ our direction be the same direction as suggested by the estimations.

It is easy to see the worst case ratio is $2\frac{\alpha^2}{\alpha-1} + 1$, but with accurate estimation, $\gamma_a = 1 + \frac{2}{\alpha-1}$. For $\alpha = 2$ these ratios are 9 and 3. $\square$