Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue

Niv Buchbinder[1], Kamal Jain[2], and Joseph (Seffi) Naor[⋆1]

[1] Computer Science Department, Technion, Haifa, Israel.
[2] Microsoft Research, Redmond, WA.

Abstract. We study the online ad-auctions problem introduced by Mehta et. al. [15]. We design a $(1 - 1/e)$-competitive (optimal) algorithm for the problem, which is based on a clean primal-dual approach, matching the competitive factor obtained in [15]. Our basic algorithm along with its analysis are very simple. Our results are based on a unified approach developed earlier for the design of online algorithms [7, 8]. In particular, the analysis uses weak duality rather than a tailor made (i.e., problem specific) potential function. We show that this approach is useful for analyzing other classical online algorithms such as ski rental and the TCP-acknowledgement problem. We are confident that the primal-dual method will prove useful in other online scenarios as well.
The primal-dual approach enables us to extend our basic ad-auctions algorithm in a straight forward manner to scenarios in which additional information is available, yielding improved worst case competitive factors. In particular, a scenario in which additional stochastic information is available to the algorithm, a scenario in which the number of interested buyers in each product is bounded by some small number d, and a general risk management framework.

1 Introduction

Maximizing the revenue of a seller in an auction has received much attention recently, and studied in many models and settings. In particular, the way search engine companies such as MSN, Google and Yahoo! maximize their revenue out of selling ad-auctions was recently studied by Mehta *et al.* [15]. In the search engine environment, advertisers link their ads to (search) keywords and provide a bid on the amount paid each time a user clicks on their ad. When users send queries to search engines, along with the (algorithmic) search results returned for each query, the search engine displays funded ads corresponding to *ad-auctions*. The ads are instantly sold, or allocated, to interested advertisers (*buyers*). The total revenue out of this fast growing market is currently billions of dollars. Thus, algorithmic ideas that can improve the allocation of the ads, even by a small percentage, are crucial. The interested reader is refered to [16] for a popular exposition of the ad-auctions problem and the work of [15].

Mehta *et al.* [15] modeled the optimal allocation of ad-auctions as a generalization of online bipartite matching [13]. There are n bidders, where each bidder i ($1 \leq i \leq n$) has a known daily budget $B(i)$. Ad-auctions, or *products*, arrive one-by-one in an online

[⋆] Work done while visiting Microsoft Research, Redmond, WA.

fashion. Upon arrival of a product, each buyer provides a bid $b(i, j)$ for buying it. The algorithm (i.e., the *seller*) then allocates the product to one of the interested buyers and this decision is irrevocable. The goal of the seller is to maximize the total revenue accrued. Mehta *et al.* [15] proposed a deterministic $(1-1/e)$-competitive algorithm for the case where the budget of each bidder is relatively large compared to the bids. This assumption is indeed realistic in the ad-auctions scenario.

1.1 Results and Techniques

We propose a simple algorithm and analysis for the online ad-auctions problem which is based on a clean primal-dual framework. The competitive ratio of our algorithm is $(1 - 1/e)$, thus matching the bounds of [15]. The primal-dual method is one of the fundamental design methodologies in the areas of approximation algorithms and combinatorial optimization. Recently, Buchbinder and Naor [7, 8] have further extended the primal-dual method and have shown its applicability to the design and analysis of online algorithms. We use the primal-dual method here for both making online decisions as well as for the analysis of the competitive factors. Moreover, we observe that several other classic online problems, e.g. ski rental and TCP acknowledgement [9, 12], for which (optimal) $e/(e - 1)$ competitive (randomized) algorithms are known, can be viewed and analyzed within the primal-dual framework, thus leading to both simpler and more general analysis. We defer the details to the full version and just sketch how to obtain the bounds. First, an $e/(e-1)$ competitive fractional solution is computed and then the solution is rounded online with no further cost, yielding an optimal randomized algorithm. This generalizes and simplifies the online framework developed in [12]. It is no coincidence that the techniques developed for the ad-auctions problem are also applicable to the ski rental and TCP acknowledgement problems; in fact, these problems are in some sense dual problems of the ad-auctions problem. Another interesting outcome of our work is a deterministic $(1 - 1/e)$-competitive *fractional* algorithm[3] for the online matching problem in bipartite graphs [13]. However, rounding with no loss the fractional solution to an integral solution, thus matching the bounds of [13], remains a challenging open problem.

We remark that in [7, 8] a primal-dual framework for online packing and covering problems is presented. This framework includes, for example, a large number of routing and load balancing problems [4, 3, 10, 8], the online set cover problem [1], as well as other problems. However, in these works only logarithmic competitive factors are achieved (which are optimal in the considered settings), while the ad-auctions problem requires much more delicate algorithms and analysis. Our analysis of the algorithms we design in this paper is very simple and uses weak duality rather than a tailor made (i.e., problem specific) potential function. We believe our results further our understanding of the primal-dual method for online algorithms and we are confident that the method will prove useful in other online scenarios as well.

Extensions. The $(1-1/e)$ competitive factor is tight for the general ad-auctions model considered by [15]. Therefore, obtaining improved competitive factors requires extending the model by relaxing certain aspects of it. The relaxations we study reveal the

[3] In fact, we show that the bound is slightly better as a function of the maximum degree.

flexibility of the primal-dual approach, thus allowing us to derive improved bounds. The algorithms developed for the different extensions (except for the bounded degree case) build very nicely on the basic ad-auctions algorithm, thus allowing us to gain more insight into the primal-dual method. We also believe that the extensions we consider result in more realistic ad-auctions models. We consider four relaxations and extensions of the basic model.

Multiple Slots. Typically, in search engines, keywords can be allocated to several advertisement slots. A slot can have several desired properties for a specific buyer, such as rank in the list of ads, size, shape, etc. We extend the basic ad-auctions algorithm to a scenario in which there are ℓ slots to which ad-auctions can be allocated. Buyers are allowed to provide *slot dependent* bids on keywords and we assume that each buyer would like to buy only a *single* slot in each round. Our basic algorithm generalizes very easily to handle this extension, yielding a competitive factor of $1 - 1/e$. Specifically, the algorithm computes in each round a maximum weight matching in a bipartite graph of slots and buyers. The proof then uses the fact that there exists an optimal primal-dual solution to the (integral) matching problem. In retrospective, our basic ad-auctions algorithm can be viewed as computing a maximum weight matching in a (degenerate) bipartite graph in which one side contains a single vertex/slot. We note that Mehta et. al. [15] also considered a multiple slots setting, but with the restriction that each bidder has the same bid for all the slots.

Incorporating Stochastic information. Suppose that it is known that a bidder is likely to spend a good fraction of its daily budget. This assumption is justified either stochastically or by experience. We want to tweak the basic allocation algorithm so that the worst case performance improves. As we tweak the algorithm it is likely that the bidder spends either a smaller or a larger fraction of its budget. Thus, we propose to tweak the algorithm gradually until a steady state is reached, i.e., no more tweaking is required. Suppose that at the steady state bidder i is likely to spent g_i fraction of its budget. In a realistic modeling of a search engine it is likely to assume that the number of times each query appears each day is more or less the same. Thus, no matter what is the exact keyword pattern, each of the advertisers spends a good fraction of its budget, say 20%. This allows us to improve the worst case competitive ratio of our basic ad-auctions algorithm. In particular, when the ratio between the bid and the budgets is small, the competitive ratio improves from $1 - 1/e$ to $1 - \frac{1-g}{e^{1-g}}$, where $g = \min_{i \in I}\{g_i\}$ is the minimum fraction of budget extracted from a buyer. As expected, the worst case competitive ratio is $(1 - 1/e)$ when $g = 0$, and it is 1 when $g = 1$.

Bounded Degree Setting. The proof of the $(1 - 1/e)$ lower bound on the competitiveness in [15] uses the fact that the number of bidders interested in a product can be unbounded and, in fact, can be as large as the total number of bidders. This assumption may not be realistic in many settings. In particular, the number of bidders interested in buying an ad for a specific query result is typically small (for most ad-auctions). Therefore, it is interesting to consider an online setting in which, for each product, the number of bidders interested in it is at most $d \ll n$. The question is whether one can take advantage of this assumption and design online algorithms with better competitive factor (better than $1 - 1/e$) in this case.

	Lower Bound	Upper Bound		Lower Bound	Upper Bound
$d = 2$	0.75	0.75	$d = 10$	0.662	0.651
$d = 3$	0.704	0.704	$d = 20$	0.648	0.641
$d = 5$	0.686	0.672	$d \to \infty$	0.6321...	0.6321 ...

Fig. 1. Summary of upper and lower bounds on the competitive ratio for certain values of d.

As a first step, we resolve this question positively in a slightly simpler setting, which we call the *allocation problem*. In the allocation problem, the seller introduces the products one-by-one and sets a fixed price $b(j)$ for each product j. Upon arrival of a product, each buyer announces whether it is interested in buying it for the set price and the seller decides (instantly) to which of the interested buyers to sell the product. We have indications that solving the more general ad-auctions problem requires overcoming a few additional obstacles. Nevertheless, achieving better competitive factors for the allocation problem is a necessary non-trivial step. We design an online algorithm with competitive ratio $C(d) = 1 - \frac{d-1}{d\left(1+\frac{1}{d-1}\right)^{d-1}}$. This factor is strictly better than $1 - 1/e$ for any value of d, and approaches $(1 - 1/e)$ from above as d goes to infinity. We also prove lower bounds for the problem that indicate that the competitive factor of our online algorithm is quite tight. Our improved bounds for certain values of d are shown in Figure 1.

Our improved competitive factors are obtained via a new approach. Our algorithm is composed of two conceptually separate phases that run simultaneously. The first phase generates online a fractional solution for the problem. A fractional solution for the problem allows the algorithm to sell each product in fractions to several buyers. This problem has a motivation of its own in case products can be divided between buyers. An example of a divisible product is the allocation of bandwidth in a communication network. This part of our algorithm that generates a fractional solution in an online fashion is somewhat counter-intuitive. In particular, a newly arrived product is not split equally between buyers who have spent the least fraction of their budget. Such an algorithm is referred to as a "water level" algorithm and it is not hard to verify that it does not improve upon the $(1-1/e)$ worst case ratio, even for small values of d. Rather, the idea is to split the product between several buyers that have **approximately** spent the same fraction of their total budget. The analysis is performed through (online) linear programming *dual fitting*: we maintain during each step of the online algorithm a dual fractional solution that bounds the optimum solution from above. We also remark that this part of the algorithm yields a competitive solution even when the prices of the products are large compared with the budgets of the buyers. As a special case, the first phase implies a $C(d)$-competitive algorithm for the online maximum fractional matching problem in bounded degree bipartite graphs [13].

The second phase consists of rounding the fractional solution (obtained in the first phase) in an online fashion. We note again that this is only a conceptual phase which is simultaneously implemented with the previous phase. This step can be easily done by using randomized rounding. However, we show how to perform the rounding deterministically by constructing a suitable potential function. The potential function is inspired by the pessimistic estimator used to derandomize the offline problem. We show that if the price of each product is small compared with the total budget of the buyer, then this rounding phase only reduces the revenue by a factor of $1 - o(1)$ compared to the revenue of the fractional solution.

Risk Management. Some researchers working in the area of ad-auctions argue that typically budgets are not strict. The reason they give is that if clicks are profitable, i.e., the bidder is expected to make more money on a click than the bid on the click, then why would a bidder want to limit its profit. Indeed, Google's Adwords program allows budget flexibility, e.g., it can overspend the budget by 20%. In fact, the arguments against daily budgets are valid for any investment choice. For example, if you consider investing ten thousand dollars in stock A and ten thousand dollars in stock B, then the expected gain for investing twenty thousand dollars in either stocks is not going to be less profitable in expectation (estimated with whatever means). Still, the common wisdom is to diversify and the reason is *risk management*. For example, a risk management tools may suggest that if a stock reaches a certain level, then execute buy/sell of this stock and/or buy/sell the corresponding call/put options.

Industry leaders are proposing risk management for ad-auctions too. The simplest form of risk management is to limit the investment. This gives us the notion of a *budget*. We consider a more complex form of real time risk management. Instead of strict budgets, we allow a bidder to specify how aggressive it wants to bid. For example, a bidder may specify that it wants to bid aggressively for the first hundred dollars of its budget. After having spent one hundred dollars, it still wants to buy ad-auctions if it gets them at, say, half of its bid. In general, a bidder has a monotonically decreasing function f of the budget spent so far specifying how aggressive it wants to bid. We normalize $f(0) = 1$, i.e., at the zero spending level the bidder is fully aggressive. If it has spent x dollars, then its next bid is scaled by a factor of $f(x)$. In Section 6 we show how to extend the primal-dual algorithm to deal with a more general scenario of real time risk management. For certain settings we also obtain better competitive factors.

1.2 Comparison to Previous Results

Maximizing the revenue of a seller in both offline and online settings has been studied extensively in many different models, e.g., [15, 2, 14, 6, 5]. The work of [15] builds on online bipartite matching [13] and online b-matching [11]. The online b-matching problem is a special case of the online ad-auctions problem in which all buyers have a budget of b, and the bids are either 0 or 1. In [11] a deterministic algorithm is given for b-matching with competitive ratio tending to $(1 - 1/e)$ (from below) as b grows.

The idea of designing online algorithms that first generate a fractional solution and then round it in an online fashion appeared implicitly in [1]. An explicit use of this idea, along with a general scheme for generating competitive online fractional solutions for packing and covering problems, appeared in [7]. Further work on primal-dual online algorithms appears in [8].

2 Preliminaries

In the online ad-auctions problem there is a set I of n buyers, each buyer i ($1 \leq i \leq n$) has a known daily budget of $B(i)$. We consider an online setting in which m products

Dual (Packing)		Primal (Covering)	
Maximize:	$\sum_{j=1}^{m} \sum_{i=1}^{n} b(i,j)y(i,j)$	Minimize :	$\sum_{i=1}^{n} B(i)x(i) + \sum_{j=1}^{m} z(j)$
Subject to:		Subject to:	
For each $1 \leq j \leq m$:	$\sum_{i=1}^{n} y(i,j) \leq 1$	For each (i,j):	$b(i,j)x(i) + z(j) \geq b(i,j)$
For each $1 \leq i \leq n$:	$\sum_{j=1}^{m} b(i,j)y(i,j) \leq B(i)$	For each i,j:	$x(i), z(j) \geq 0$
For each i,j:	$y(i,j) \geq 0$		

Fig. 2. The fractional ad-auctions problem (the dual) and the corresponding primal problem

arrive one-by-one in an online fashion. Let M denote the set of all the products. The bid of buyer i on product j (which states the amount of money it is willing to pay for the item) is $b(i,j)$. The online algorithm can *allocate* (or *sell*) the product to any one of the buyers. We distinguish between *integral* and *fractional* allocations. In an integral allocation, a product can only be allocated to a single buyer. In a fractional allocation, products can be fractionally allocated to several buyers, however, for each product, the sum of the fractions allocated to buyers cannot exceed 1. The revenue received from each buyer is defined to be the minimum between the sum of the costs of the products allocated to a buyer (times the fraction allocated) and the total budget of the buyer. That is, buyers can never be charged by more than their total budget. The objective is to maximize the total revenue of the seller. Let $R_{\max} = \max_{i \in I, j \in M}\{\frac{b(i,j)}{B(i)}\}$ be the maximum ratio between a bid of any buyer and its total budget.

A linear programming formulation of the fractional (offline) ad-auctions problem appears in Figure 2. Let $y(i,j)$ denote the fraction of product j allocated to buyer i. The objective function is maximizing the total revenue. The first set of constraints guarantees that the sum of the fractions of each product is at most 1. The second set of constraints guarantees that each buyer does not spend more than its budget. In the primal problem there is a variable $x(i)$ for each buyer i and a variable $z(j)$ for each product j. For all pairs (i,j) the constraint $b(i,j)x(i) + z(j) \geq b(i,j)$ needs to be satisfied.

3 The Basic Primal-Dual Online Algorithm

The basic algorithm for the online ad-auctions produces primal and dual solutions to the linear programs in Figure 2.

Allocation Algorithm: Initially $\forall i \;\; x(i) \leftarrow 0$.
Upon arrival of a new product j allocate the product to the buyer i that maximizes $b(i,j)(1 - x(i))$. If $x(i) \geq 1$ then do nothing. Otherwise:

1. Charge the buyer the minimum between $b(i,j)$ and its remaining budget and set $y(i,j) \leftarrow 1$
2. $z(j) \leftarrow b(i,j)(1 - x(i))$
3. $x(i) \leftarrow x(i)\left(1 + \frac{b(i,j)}{B(i)}\right) + \frac{b(i,j)}{(c-1) \cdot B(i)}$ (c is determined later).

The intuition behind the algorithm is the following. If the competitive ratio we are aiming for is $1 - 1/c$, then we need to guarantee that in each iteration the change in the primal cost is at most $1 + 1/(c-1)$ the change in the dual profit. The value of c is then maximized such that both the primal and the dual solutions remain feasible.

Theorem 1. *The allocation algorithm is* $(1-1/c)\,(1-R_{\max})$*-competitive, where* $c = (1+R_{\max})^{\frac{1}{R_{\max}}}$*. When* $R_{\max} \to 0$ *the competitive ratio tends to* $(1-1/e)$*.*

Proof. We prove three simple claims:

1. The algorithm produces a primal feasible solution.
2. In each iteration: (*change in primal objective function*)/ (*change in dual objective function*) $\leq 1 + \frac{1}{c-1}$.
3. The algorithm produces an almost feasible dual solution.

Proof of (1)**:** Consider a primal constraint corresponding to buyer i and product j. If $x(i) \geq 1$ then the primal constraint is satisfied. Otherwise, the algorithm allocates the product to the buyer i' for which $b(i',j)(1-x(i'))$ is maximized. Setting $z(j) = b(i',j)(1-x(i'))$ guarantees that the constraint is satisfied for all (i,j). Subsequent increases of the variables $x(i)$'s cannot make the solution infeasible.

Proof of (2)**:** Whenever the algorithm updates the primal and dual solutions, the change in the dual profit is $b(i,j)$. (Note that even if the remaining budget of buyer i to which product j is allocated is less than its bid $b(i,j)$, variable $y(i,j)$ is still set to 1.) The change in the primal cost is:

$$B(i)\Delta x(i) + z(j) = \left(b(i,j)x(i) + \frac{b(i,j)}{c-1}\right) + b(i,j)(1-x(i)) = b(i,j)\left(1 + \frac{1}{c-1}\right).$$

Proof of (3)**:** The algorithm never updates the dual solution for buyers satisfying $x(i) \geq 1$. We prove that for any buyer i, when $\sum_{j\in M} b(i,j)y(i,j) \geq B(i)$, then $x(i) \geq 1$. This is done by proving that:

$$x(i) \geq \frac{1}{c-1}\left(c^{\frac{\sum_{j\in M} b(i,j)y(i,j)}{B(i)}} - 1\right). \tag{1}$$

Thus, whenever $\sum_{j\in M} b(i,j)y(i,j) \geq B(i)$, we get that $x(i) \geq 1$. We prove (1) by induction on the (relevant) iterations of the algorithm. Initially, this assumption is trivially true. We are only concerned with iterations in which a product, say k, is sold to buyer i. In such an iteration we get that:

$$\begin{aligned}
x(i)_{\text{end}} &= x(i)_{\text{start}} \cdot \left(1 + \frac{b(i,k)}{B(i)}\right) + \frac{b(i,k)}{(c-1)\cdot B(i)} \\
&\geq \frac{1}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{k\}} b(i,j)y(i,j)}{B(i)}} - 1\right]\cdot\left(1 + \frac{b(i,k)}{B(i)}\right) + \frac{b(i,k)}{(c-1)\cdot B(i)} \qquad (2) \\
&= \frac{1}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{k\}} b(i,j)y(i,j)}{B(i)}} \cdot \left(1 + \frac{b(i,k)}{B(i)}\right) - 1\right] \\
&\geq \frac{1}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{k\}} b(i,j)y(i,j)}{B(i)}} \cdot c^{\left(\frac{b(i,k)}{B(i)}\right)} - 1\right] = \frac{1}{c-1}\left[c^{\frac{\sum_{j\in M} b(i,j)y(i,j)}{B(i)}} - 1\right] \qquad (3)
\end{aligned}$$

where Inequality (2) follows from the induction hypothesis, and Inequality (3) follows since, for any $0 \leq x \leq y \leq 1$, $\frac{\ln(1+x)}{x} \geq \frac{\ln(1+y)}{y}$. Note that when $\frac{b(i,k)}{B(i)} = R_{\max}$

then Inequality 3 holds with equality. This is the reason why we chose the value c to be $(1+R_{\max})^{\frac{1}{R_{\max}}}$. Thus, it follows that whenever the sum of charges to a buyer exceeds the budget, we stop charging this buyer. Hence, there can be at most one iteration in which a buyer is charged by less than $b(i,j)$. Therefore, for each buyer i: $\sum_{j\in M} b(i,j)y(i,j) \le B(i) + \max_{j\in M}\{b(i,j)\}$, and thus the profit extracted from buyer i is at least:

$$\left[\sum_{j\in M} b(i,j)y(i,j)\right]\frac{B(i)}{B(i)+\max_{j\in M}\{b(i,j)\}} \ge \left[\sum_{j\in M} b(i,j)y(i,j)\right](1-R_{\max}).$$

By the second claim the dual it at least $1-1/c$ times the primal, and thus (by weak duality) we conclude that the competitive ratio of the algorithm is $(1-1/c)\,(1-R_{\max})$.

3.1 Multiple Slots

In this section we show how to extend the algorithm in a very elegant way to sell different advertisement slots in each round. Suppose there are ℓ slots to which ad-auctions can be allocated and suppose that buyers are allowed to provide bids on keywords which are slot dependent. Denote the bid of buyer i on keyword j and slot k by $b(i,j,k)$. The restriction is that an (integral) allocation of a keyword to two different slots cannot be sold to the same buyer. The linear programming formulation of the problem is in Figure 3. Note that the algorithm does not update the variables $z(\cdot)$ and $s(\cdot)$ explicitly. These variables are only used for the purpose of analysis, and are updated conceptually in the proof using the strong duality theorem. The algorithm for the online ad-auctions problem is as follows.

Allocation Algorithm: Initially, $\forall i,\;\; x(i) \leftarrow 0$. Upon arrival of a new product j:

1. Generate a bipartite graph H: n buyers on one side and ℓ slots on the other side. Edge $(i,k)\in H$ has weight $b(i,j,k)(1-x(i))$.
2. Find a maximum weight (integral) matching in H, i.e., an assignment to the variables $y(i,j,k)$.
3. Charge buyer i the minimum between $\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)$ and its remaining budget.
4. For each buyer i, if there exists slot k for which $y(i,j,k)>0$:

$$x(i) \leftarrow x(i)\left(1+\frac{b(i,j,k)y(i,j,k)}{B(i)}\right)+\frac{b(i,j,k)y(i,j,k)}{(c-1)\cdot B(i)}$$

Theorem 2. *The algorithm is $(1-1/c)\,(1-R_{\max})$-competitive, where c tends to e when $R_{\max}\to 0$.*

4 Incorporating Stochastic information

In this Section we improve the worst case competitive ratio when additional stochastic information is available. We assume that stochastically or with historical experience we

Dual (Packing)	
Maximize:	$\sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{\ell=1}^{k} b(i,j,\ell)y(i,j,\ell)$
Subject to:	
$\forall 1 \le j \le m, 1 \le k \le \ell$:	$\sum_{i=1}^{n} y(i,j,k) \le 1$
$\forall 1 \le i \le n$:	$\sum_{j=1}^{m}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k) \le B(i)$
$\forall 1 \le j \le m, 1 \le i \le n$:	$\sum_{k=1}^{\ell} y(i,j,k) \le 1$
Primal (Covering)	
Minimize :	$\sum_{i=1}^{n} B(i)x(i) + \sum_{j=1}^{m}\sum_{k=1}^{\ell} z(j,k) + \sum_{i=1}^{n}\sum_{j=1}^{m} s(i,j)$
Subject to:	
$\forall i,j,k$:	$b(i,j,k)x(i) + z(j,k) + s(i,j) \ge b(i,j,k)$

Fig. 3. The fractional multi-slot problem (the dual) and the corresponding primal problem

know that, a bidder i is likely to spent a good fraction of her budget. We want to tweak the algorithm so that the algorithm's worst case performance improves. As we tweak the algorithm it is likely that the bidder may spent more or less fraction of her budget. So we propose to tweak the algorithm gradually until some steady state is reached, i.e., no more tweaking is required. Suppose at the steady state, buyer i is likely to spent a good fraction of his budget. Let $0 \le g_i \le 1$ be a lower bound on the fraction of the budget buyer i is going to spend. We show that having this additional information allows us to improve the worst case competitive ratio to $1 - \frac{1-g}{e^{1-g}}$, where $g = \min_{i \in I}\{g_i\}$ is the minimal fraction of budget extracted from a buyer.

The main idea behind the algorithm is that if a buyer is known to spend at least g_i fraction of his budget, then it means that the corresponding primal variable $x(i)$ will be large at the end. Thus, in order to make the primal constraint feasible, the value of $z(j)$ can be made smaller. This, in turn, gives us additional "money" that can be used to increase $x(i)$ faster. The tradeoff we have is on the value that $x(i)$ is going to be once the buyer spent g_i fraction of his budget. This value is denoted by $x_s(i)$ and we choose it so that after the buyer has spent g_i fraction of its budget, $x(i) = x_s(i)$, and after having extracting all its budget, $x(i) = 1$. In addition, we need the change in the primal cost to be the same with respect to the dual profit in iterations where we sell the product to a buyer i who has not yet spent the threshold of g_i of his budget. The optimal choice of $x_s(i)$ turns out to be $\frac{g_i}{c^{1-g_i}-(1-g_i)}$, and the growth function of the primal variable $x(i)$, as a function of the fraction of the budget spent, should be linear until the buyer has spent a g_i fraction of his budget, and exponential from that point on. The modified algorithm is the following:

Allocation Algorithm: Initially $\forall i \;\; x(i) \leftarrow 0$. Upon arrival of a new product j Allocate the product to the buyer i that maximizes $b(i,j)(1 - \max\{x(i), x_s(i)\})$, where $x_s(i) = \frac{g_i}{c^{1-g_i}-(1-g_i)}$. If $x(i) \ge 1$ then do nothing. Otherwise:

1. Charge the buyer the minimum between $b(i,j)$ and its remaining budget
2. $z(j) \leftarrow b(i,j)(1 - \max\{x(i), x_s(i)\})$
3. $x(i) \leftarrow x(i) + \max\{x(i), x_s(i)\}\frac{b(i,j)}{B(i)} + \frac{b(i,j)}{B(i)}\frac{1-g_i}{c^{1-g_i}-(1-g_i)}$ (c is determined later).

Theorem 3. *If each buyer spends at least g_i fraction of its budget, then the algorithm is:* $\left(1 - \frac{1-g}{c^{1-g}}\right)(1 - R_{\max})$*-competitive, where* $c = (1 + R_{\max})^{\frac{1}{R_{\max}}}$.

5 Bounded Degree Setting

In this section we improve on the competitive ratio under the assumption that the number of buyers interested in each product is small compared with the total number of buyers. To do so, we design a modified primal-dual based algorithm. The algorithm only works in the case of a simpler setting (which is still of interest) called the *allocation problem*. Still, this construction turns out to be non-trivial and gives us additional useful insight into the primal-dual approach. In the allocation problem, a seller is interested in selling products to a group of buyers, where buyer i has budget $B(i)$. The seller introduces the products one-by-one and sets a fixed price $b(j)$ for each product j. Each buyer then announces to the seller (upon arrival of a product) whether it is interested in buying the current product for the set price. The seller then decides (instantly) to which of the interested buyers to sell the product. For each product j let $S(j)$ be the set of interested buyers. We assume that there exists an upper bound d such that for each product j, $|S(j)| \leq d$.

The main idea is to divide the buyers into *levels* according to the fraction of the budget that they have spent. For $0 \leq k \leq d$, let $L(k)$ be the set of buyers that have spent at least a fraction of $\frac{k}{d}$ and less than a fraction of $\frac{k+1}{d}$ of their budget (buyers in level d exhausted their budget). We refer to each $L(k)$ as level k and say that it is *nonempty* if it contains buyers. We design an algorithm for the online allocation problem using two conceptual steps. First, we design an algorithm that is allowed to allocate each product in fractions. We bound the competitive ratio of this algorithm with respect to the optimal fractional solution for the problem. We then show how to deterministically produce an integral solution that allocates each product to a single buyer. We prove that when the prices of the products are small compared to the total budget, the loss of revenue in this step is at most an $o(1)$ with respect to the fractional solution.

Our fractional allocation algorithm is somewhat counter-intuitive. In particular, the product is not split equally between buyers that spent the least fraction of their budget[4], but rather to several buyers that have approximately spent the same fraction of their total budget. The formal description of the algorithm is the following:

Allocation Algorithm: Upon arrival of a new product j allocate the product to the buyers according to the following rules:

- Allocate the product equally and continuously between interested buyers in the lowest non empty level that contain buyers from $S(j)$. If during the allocation some of the buyers moved to a higher level, then continue to allocate the product equally only among the buyers in the lowest level.
- If all interested buyers in the lowest level moved to a higher level, then allocate the remaining fraction of the product equally and continuously between the buyers in the new lowest level. If all interested buyers have exhausted their budget, then stop allocating the remaining fraction of the product.

[4] Such an algorithm is usually called a "water level" algorithm.

The idea of the analysis is to find the best tradeoff function, f_d, that relates the value of each primal variable to the value of its corresponding dual constraint. It turns out that the best function is a piecewise linear function that consists of d linear segments. As d grows, the function approximates the exponential function $f_{(d=\infty)}(x) = \frac{e^x-1}{e-1}$. Due to lack of space we defer the full analysis to the full version of the paper and just state the main theorem we proved.

Theorem 4. *The allocation algorithm is $C(d)$-competitive with respect to the optimal offline fractional solution, where: $C(d) = 1 - \frac{d-1}{d\left(1+\frac{1}{d-1}\right)^{d-1}}$.*

As stated we prove that when the prices of the products are small compared to the total budget, we may transform the fractional allocation to an integral allocation with only a small loss of revenue. We do so, by introducing another potential function that is used to make the rounding decisions. We defer the description of this part to the full version of the paper.

Lower Bounds. As we stated earlier, the standard lower bound example makes use of products with large number of interested buyers. Though, the same example when restricted to bounded degree d gives quite tight bounds. Inspecting this bound more accurately we can prove the following lower bound for any value d. Figure 1 provides the lower bounds for several sample values of d.

Lemma 1. *For any value d: $C(d) \leq 1 - \frac{k-kH(d)+\sum_{i=1}^{k} H(d-i)}{d}$, where $H(\cdot)$ is the harmonic number, and k is the largest value for which $H(d) - H(d-k) \leq 1$*

Our bound is only tight for $d = 2$. We can derive better tailor-made lower bounds for specific values of d. In particular it is not hard to show that our algorithm is optimal for $d = 3$. We defer this result to the full version of the paper.

6 Risk Management

We extend our basic ad-auctions algorithm to handle a more general setting of real time risk management. Here each buyer has a monotonically decreasing function f of budget spent, specifying how aggressive it wants to bid. We normalize $f(0) = 1$, i.e., at the zero spending level it is fully aggressive. If it has spent x dollars then its next bid is scaled by a factor of $f(x)$. Note that since f is a monotonically decreasing function, the revenue obtained by allocating buyer i a set of items is a concave function of $\sum_j b(i,j)y(i,j)$.

Since we are interested in solving this problem integrally we assume the revenue function is piecewise linear. Let R_i be the revenue function of buyer i. Let r_i be the number of pieces of the function R_i. We define for each buyer $(r_i - 1)$ different budgets $B(i,r)$, defining the amount of money spent in each "aggression" level. When the buyer spends money from budget $B(i,r)$, the aggression ratio is $a(i,r) \leq 1$ ($a(i,1) = 1$ for each buyer i). We then define a new linear program with variables $y(i,j,k)$ indicating that item j is sold to buyer i using the kth budget. Note that the ad-auctions problem considered earlier is actually this generalized problem with two pieces. In some scenarios it is likely to assume that each buyer has a lowest "aggression" level that is strictly more than zero. For instance, a buyer is always willing to buy an item if he only needs to pay 10% of its value (as estimated by the buyer's bid). Our modified algorithm for this more general setting takes advantage of this fact to improve the worst case competitive ratio. In particular, let $a_{\min} = \min_{i=1}^{n}\{a(i, r_i)\}$ be the minimum "aggression" level of the buyers, then the competitive factor of the algorithm is $\frac{e-1}{e-a_{\min}}$. If the minimum level is only 10% (0.1), for example, the competitive ratio is 0.656, compared with $0.632 \approx 1 - 1/e$ of the basic ad-auctions algorithm. Let $R_{\max} = \max_{i \in I, j \in M, 1 \leq r \leq r_i - 1}\{\frac{a(i,r)b(i,j)}{B(i,r)}\}$ be the maximum ratio between a charge to a budget and the total budget. The modified linear program for the more general risk management setting along with our modified algorithm are deferred to the full version.

Theorem 5. *The algorithm is* $\left(\frac{c-1}{c-a_{\min}}\right)(1 - R_{\max})$*-competitive, where* c *tends to* e *when* $R_{\max} \to 0$.

Acknowledgements: We thank Allan Borodin for pointing to us the mutiple slot setting.

References

1. N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. The online set cover problem. In *Proceedings of the 35th STOC*, pp. 100-105, 2003.
2. N. Andelman and Y. Mansour. Auctions with budget constraints. In *Proc. of the 9th Scandinavian Workshop on Algorithm Theory*, pp. 26-38, 2004.
3. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *J. ACM*, 44(3):486-504, 1997.
4. B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *Proc. of the 34th Annual Symposium on Foundations of Computer Science*, pp. 32-40, 1993.
5. A. Blum and J. Hartline. Near-optimal online auctions. 2005.
6. C. Borgs, J. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *Proc. of the 6th EC*, pp. 44-51, 2005.
7. N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. In *Proc. of the 13th Annual European Symposium on Algorithms*, pp. 689-701, 2005.
8. N. Buchbinder and J. Naor. A primal-dual approach to online routing and packing. In *Proc. of the 47th Annual Symposium on Foundations of Computer Science*, 293-204, 2006.
9. D. R. Dooly, S. Goldman, and S. D. Scott. On-line analysis of the TCP acknowledgement delay problem. *Journal of the ACM*, 48:243-273, 2001.
10. A. Goel, A. Meyerson, and S. A. Plotkin. Combining fairness with throughput: Online routing with multiple objectives. *J. Comput. Syst. Sci.*, 63(1):62-79, 2001.
11. B. Kalyanasundaram and K. R. Pruhs. An optimal deterministic algorithm for online b -matching. *Theoretical Computer Science*, 233(1-2):319-325, 2000.
12. A. R. Karlin, C. Kenyon, and D. Randall. Dynamic TCP acknowledgement and other stories about e/(e-1). In *Proc. of the 33rd Symposium on Theory of Computing*, pp. 502-509, 2001.
13. R. Karp, U. Vazirani, and V. Vazirani. An optimal algorithm for online bipartite matching. In *In Proceedings of the 22nd Annual Symposium on Theory of Computing*, pp. 352-358, 1990.
14. M. Mahdian and A. Saberi. Multi-unit auctions with unknown supply. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pp. 243-249, 2006.
15. A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized on-line matching. In *Proc. of the 46th IEEE Symp. on Foundations of Computer Science*, 264-273, 2005.
16. S. Robinson. Computer scientists optimize innovative ad auction. *SIAM News*, 38:243-273, 2005.