Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue

Niv Buchbinder [*] Kamal Jain [†] Joseph (Seffi) Naor [‡]

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 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.

[*]Computer Science Department, Technion, Haifa 32000, Israel. E-mail: *nivb@cs.technion.ac.il*

[†]Microsoft Research, Redmond, WA. E-mail: *kamalj@microsoft.com*

[‡]Microsoft Research, Redmond, WA. On leave from the CS Dept., Technion, Haifa, Israel. E-mail: *naor@cs.technion.ac.il*

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. Our bounds for the two problems are optimal and are obtained as follows. 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[1] 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 an 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.

1.1.1 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 *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

[1] In fact, in Section 5 we show that the bound is slightly better as a function of the maximum degree.

	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 ...

Figure 1: Summary of upper and lower bounds on the competitive ratios for certain values of d.

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.

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 dollars, 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].

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 $1 \le j \le m$:	$\sum_{i=1}^{n} y(i,j) \le 1$	For each (i,j):	$b(i,j)x(i) + z(j) \ge b(i,j)$
For each $1 \le i \le n$:	$\sum_{j=1}^{m} b(i,j)y(i,j) \le B(i)$	For each i,j:	$x(i), z(j) \ge 0$
For each i,j:	$y(i,j) \ge 0$		

Figure 2: The fractional ad-auctions problem (the dual) and the corresponding primal problem

2 Preliminaries

In the online ad-auctions problem there is a set I of n buyers, each buyer i ($1 \le i \le n$) has a known daily budget of $B(i)$. We consider an online setting in which m products 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 of the seller. 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, variable $z(j)$ for each product j, and for all pairs (i,j) the constraint $b(i,j)x(i) + z(j) \ge b(i,j)$ needs to be satisfied.

3 The Basic Primal-Dual Online Algorithm

We present here the basic algorithm for the online ad-auctions problem and analyze it via the primal dual method. The extensions to ski rental and TCP acknowledgement problems are presented in Section 7.

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

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 $x(i) \ge 1$ 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).

Theorem 3.1. *The algorithm is* $(1-1/c)(1-R_{\max})$*-competitive, where* $c = (1+R_{\max})^{\frac{1}{R_{\max}}}$*. When* $R_{\max}$ *tends to 0 the competitive ratio of the algorithm 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).
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:

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

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: $\sum_{j\in M} b(i,j)y(i,j) \leq B(i) + \max_{j\in M}\{b(i,j)\}$, and thus the profit extracted from buyer i is at least:

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

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 $(1-1/c)(1-R_{\max})$. □

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:	
$\forall i,j,k$:	$b(i,j,k)x(i) + z(j,k) + s(i,j) \ge b(i,j,k)$

Figure 3: The fractional multi-slot ad-auction problem (the dual) and the corresponding primal problem

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.

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 $(i,k) \in H$, if $b(i,j,k)(1-x(i)) > 0$; (i,k) 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 $y(i,j,k) > 0$:

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

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.

Theorem 3.2 (Proof in Appendix B). *The 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 of the algorithm tends to* $(1-1/e)$.

4 Incorporating Stochastic information

In this Section we improve the worst case competitive ratio when additional stochastic information is available. We assume that each bidder is likely to spend 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 $x(i) \geq 1$ 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. $z(j) \leftarrow b(i,j)(1 - \max\{x(i), x_s(i)\})$
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 4.1. *(***Proof is in Appendix B***) 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 $B(i)$. Define $B_{\min} = \min_i B(i)$. 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. In Section 5.1 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 in Section 5.2 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.

5.1 Obtaining Competitive Fractional Solution

In this section we describe a simple algorithm that produces a fractional solution to the allocation problem. Note that our allocation algorithm is somewhat counter-intuitive. In particular, the product is not split equally between buyers that spent the least fraction of their budget, but rather to several buyers that have approximately spent the same fraction of their total budget.

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

1. Allocate the product equally and continuously between interested buyers in the lowest non empty level that contain buyers from $S(j)$.
2. 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.
3. If all interested buyers in the lowest level moved to a higher level, then start allocating the remaining fraction of the product equally and continuously between the buyers in the new lowest level that contain buyers from $S(j)$.
4. If all interested buyers have exhausted their budget, then stop allocating the remaining fraction of the product.

As a first step towards the analysis of the algorithm, define the following potential function f_d for any parameter d over the interval $[0, 1]$. The function f_d is piecewise linear and consists of d linear segments. In order to define the segments we first define a geometric sequence a_t ($1 \le t \le d$) inductively as follows:

$$a_1 = \frac{1}{d\left(1+\frac{1}{d-1}\right)^{d-1} - (d-1)}, \qquad \ldots \qquad , a_t = a_1 \cdot \left(1 + \frac{1}{d-1}\right)^{t-1}.$$

Thus, the sequence a_t is a geometric sequence and we only consider the first d elements in the sequence. The potential function f_d is defined for any $0 \le j \le d$ to be $f_d(\frac{j}{d}) \triangleq \sum_{t=1}^{j} a_t$. A simple calculation yields the following, for any j, $1 \le j \le d$:

$$f_d\left(\frac{j}{d}\right) = \sum_{i=1}^{j} a_i = a_1 \cdot \frac{\left(1+\frac{1}{d-1}\right)^j - 1}{(1+\frac{1}{d-1}) - 1} = a_1 \cdot \left[d\left(1 + \frac{1}{d-1}\right)^{j-1} - (d-1)\right].$$

In particular, setting $j = d$, we get $f_d\left(\frac{d}{d}\right) = 1$. As d grows, the potential function f_d actually approximates the exponential function $f_{(d=\infty)}(x) = \frac{e^x - 1}{e-1}$. This piecewise linear approximation allows us to analyze more accurately the algorithm and obtain better competitive factors. The function f_d for $d = 2$, $d = 3$, and for d tending to ∞ appears in Figure 4. Next, we use the potential function to prove that the allocation algorithm has the desired competitive factor. Figure 1 provides the competitive ratio for several sample values of d.

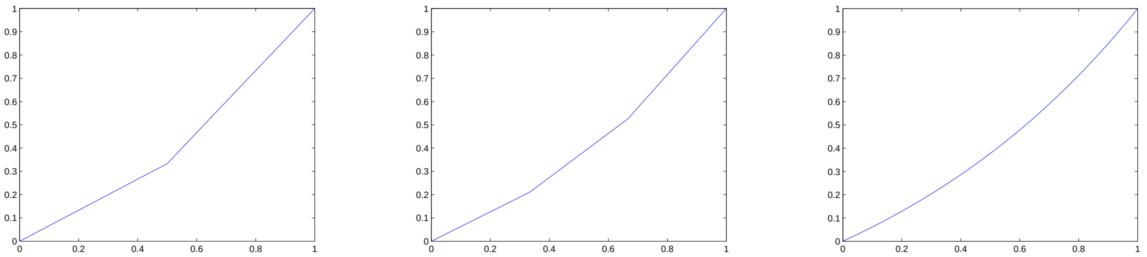
Theorem 5.1. ***(Proof is in Appendix B.)*** *The allocation algorithm is $C(d)$-competitive with respect to the optimal offline fractional solution, where:* $C(d) = 1 - \frac{d-1}{d\left(1+\frac{1}{d-1}\right)^{d-1}}$.

The theorem implies a $C(d)$-competitive fractional algorithm for the online bipartite matching problem [13] in bounded degree graphs.

5.2 Deterministic Rounding of the Fractional Solution

In this section we show how to deterministically round the fractional solution in an online fashion so as to get a feasible integral solution with only a negligible loss in revenue. To this end, we introduce the following potential function $\Phi = \Phi_1 + \Phi_2$. Initially, $\chi(i, j) = 0$ for all i and j.

$$\Phi_1 = \frac{1}{2n}\sum_{i=1}^{n} \exp\left\{\sum_{j \mid i \in S(j)} \ln\left(1 + \sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j)\right)\chi(i,j) - \sum_{j \mid i\in S(j)} \sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j) y(i,j)\right\}.$$



$d = 2$ $\quad$ $d = 3$ $\quad$ $d \to \infty$

Figure 4: The function f_d for $d = 2$ and $d = 3$. The y middle value for $d = 2$ is $1/3$. The y middle values for $d = 3$ are $4/19$ and $10/19$.

$$\Phi_2 = \frac{1}{2} \exp \left\{ \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} \frac{1}{\sqrt{B_{\min}}} b(j) y(i,j) + \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} \ln \left(1 - \frac{b(j)}{\sqrt{B_{\min}}} \right) \chi(i,j) \right\}.$$

Remark 5.2. *Since products arrive online, the potential function Φ contains only variables that correspond to products that have already arrived. Thus, a slightly better notation may be $\Phi(k)$, denoting the potential following the arrival of the kth product. However, the current notation simplifies the discussion and proofs.*

The integral allocation algorithm is very simple.

1. Run the fractional allocation algorithm.

2. In each iteration j, if there exists a buyer i such that setting $\chi(i,j) = 1$ does not increase the potential function Φ, then set $\chi(i,j) = 1$ (breaking ties arbitrarily).

3. If $\chi(i,j) = 1$ and the (residual) budget of buyer i is greater than or equal to $b(j)$, then allocate the product to buyer i and decrease its budget by $b(j)$.

Note that in case the residual budget of buyer i is not sufficient for buying product j, then we still set $\chi(i,j) = 1$. (This happens even if the residual budget is zero.) Thus, if $\chi(i,j) = 1$ then product j is said to be *virtually allocated* to i. Note that in case the budget of buyer i suffices for buying product j, then j is both allocated and virtually allocated to i.

Intuitively, the second term of the potential function, Φ_2, ensures that the total sum of the prices of virtually allocated products is close to the sum of prices of products that were allocated by the fractional solution. The main concern, then, is that not all the products that are virtually allocated are also allocated, and thus, the revenue may be smaller. To this end, the first term of the potential function ensures that the total price of products that are virtually allocated to each buyer is at most $1 + o(1)$ times its budget. Thus, in total most of the products that are virtually allocated are also allocated. This analysis is done formally in the following. We start by proving some properties of the potential function Φ.

Lemma 5.3. *The potential function Φ satisfies the following:*

1. *Initially, $\Phi \leq 1$; throughout the algorithm, $\Phi > 0$.*

2. *In each iteration of the integral allocation algorithm, Φ does not increase, i.e., Φ is monotonically non-increasing throughout the algorithm.*

The next theorem states that the integral allocation algorithm only reduces the total revenue by a fraction of $o(1)$. This is true only if the total budget is large compared with the prices of the products.

Theorem 5.4. *(***Proof is in Appendix B.***) Let $b_{\max}$ denote the maximum cost of a product. The revenue of the integral allocation algorithm is at least $1-o(1)$ times the revenue of the fractional solution, provided that:*

$$(1+b_{\max}) \cdot \sqrt{\frac{\ln 2n}{B_{\min}}} = o(1) \tag{4}$$

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. The proofs of the lemmas appear in Appendix B.

Lemma 5.5. *For any value d:*

$$C(d) \le 1 - \frac{k - kH(d) + \sum_{i=1}^{k} H(d-i)}{d},$$

where $H(\cdot)$ is the harmonic number, and k is the largest value for which $H(d) - H(d-k) \le 1$

Our bound is only tight for $d = 2$. We can derive better tailor-made bounds for specific values of d. The next lemma shows that our algorithm achieves an optimal competitive ratio for $d = 3$.

Lemma 5.6.

$$C(3) \le \frac{19}{27} = 0.704$$

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 $a(i,r) \le 1$ ($a(i,1) = 1$ 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 $R_{\max} = \max_{i\in I, j\in M, 1\le r\le r_i-1}\{\frac{a(i,r)b(i,j)}{B(i,r)}\}$ 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 and the proof of Theorem 6.1 appear in Appendix B.

Theorem 6.1. *The algorithm is $\left(\frac{c-1}{c-a_{\min}}\right)(1-R_{\max})$-competitive, where $c = (1+R_{\max})^{\frac{1}{R_{\max}}}$. When $R_{\max} \to 0$ the competitive ratio of the algorithm tends to $\frac{e-1}{e-a_{\min}}$.*

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

Figure 5: The fractional ski problem (the primal) and the corresponding dual problem

7 Ski Rental and Dynamic TCP Acknowledgement Problems

In this section we show how to obtain optimal fractional algorithms, as well as randomized integral algorithms, using the online primal-dual technique developed in the paper for the well known ski rental and dynamic TCP acknowledgement problems. Both problems were considered previously, e.g. [12]. We show a very simple algorithm and simple analysis that uses weak duality. Both problems are, in fact, minimization problems that are very close to the dual problem of the ad-auctions problem. Due to lack of space the dynamic TCP acknowledgement problem appears in Appendix A. We believe that our approach to the design and analysis of online algorithm via the primal-dual method is quite general and will find additional applications in the future, in both minimization and maximization online problems.

The Ski Rental Problem: The classic ski rental problem is the following. A customer arrives at a ski resort, where renting skis costs \$1 per day, while buying skis costs $\$B$. The unknown factor is the number of skiing days left before the snow melts. This is the customer's last vacation, so the goal is to minimize the total expenses. We develop both fractional algorithms, as well as randomized integral algorithms, using a primal-dual approach. The offline problem (with m skiing days) can be formulated by the simple covering linear system in Figure 5. Variable x is set to 1 if we decide to buy the skis. For each day j, variable $z(j)$ is set to 1 if we decide to rent the skis on that day. The constraints guarantee that on each day we either rent skis or buy them. The dual system is also extremely simple and consists of variables $y(j)$ corresponding to each day j. In the fractional version of the problem it is required that the sum of the fractions corresponding to renting and buying is at least 1 on each day. In the online setting new primal constraints (days) arrive one by one. Upon arrival, each primal constraints should be satisfied, and we demand that the primal variables can only be increased. We are now ready to present the online algorithm.

Initially, $x \leftarrow 0$. Each new day (jth new constraint), if $x < 1$:

1. $z(j) \leftarrow 1 - x$ and $x \leftarrow x\left(1 + \frac{1}{B}\right) + \frac{1}{(c-1) \cdot B}$. (The value of c is determined later.)
2. $y(j) \leftarrow 1$.

The analysis is extremely simple. We need to show: (i) the primal and dual solutions are feasible; (ii) in each iteration, the ratio between the change in the primal and dual objective functions is bounded by $(1 + 1/(c-1))$. This will prove that the algorithm is $(1 + 1/(c-1))$-competitive.

First, it is easy to see that the primal solution we produce on each day is feasible. Second, if $x < 1$, the dual objective function increases by 1, and the increase in the primal objective function is $B\Delta x + z(j) = x + 1/(c-1) + 1 - x = 1 + 1/(c-1)$, thus the ratio is $(1 + 1/(c-1))$. Third, to show feasibility of the dual solution, we need to show that $\sum_{j=1}^{m} y(j) \leq B$. We prove that $x \geq 1$ after at most B days of ski. Variable x is the sum of a geometric sequence in which $a_1 = 1/((c-1)B)$ and $q = 1 + 1/B$. Thus, after B days of ski, $x = \frac{\left(1+\frac{1}{B}\right)^{B} - 1}{c-1}$, implying that $c \leq (1 + \frac{1}{B})^B$. Thus, the competitive ratio is $(1 - 1/e)$ when $B \gg 1$. We note that using the techniques in Section 5, it is possible to improve the competitive ratio to $C(B) = 1 - \frac{B-1}{B\left(1+\frac{1}{B-1}\right)^{B-1}}$.

In order to get a randomized integral solution, we arrange the increments of x on the interval $[0, 1]$ and choose uniformly in random a number in $[0, 1]$. We buy on the day corresponding to the increment of x to which the random number belongs. It can be shown that the probability of buying on the jth day is exactly the change in the value of x on the jth day and the probability of renting the skis on the jth day is exactly $z(j)$. Thus, the expected cost of the randomized algorithm is the same as the cost of the fractional algorithm.

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 annual ACM Symposium on the Theory of Computation*, pages 100–105, 2003.

[2] N. Andelman and Y. Mansour. Auctions with budget constraints. In *SWAT 2004: 9th Scandinavian Workshop on Algorithm Theory*, pages 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 34th FOCS*, pages 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 ACM conference on Electronic commerce*, pages 44–51, 2005.

[7] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. In *13th Annual European Symposium on Algorithms - ESA 2005*, 2005.

[8] N. Buchbinder and J. Naor. A primal-dual approach to online routing and packing. In *To appear in 47th Annual IEEE Symposium on Foundations of Computer Science*, 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 *ACM Symposium on Theory of Computing*, pages 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 ACM Symposium on Theory of Computing*, pages 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*, pages 243–249, 2006.

[15] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized on-line matching. In *Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 264–273, 2005.

[16] S. Robinson. Computer scientists optimize innovative ad auction. *SIAM News*, 38:243–273, 2005.

Dual (Packing)		Primal (Covering)	
Maximize:	$\sum_{j\in M}\sum_{t\mid t\geq t(j)} y(j,t)$	Minimize :	$\sum_{t\in T} x_t + \sum_{j\in M}\sum_{t\mid t\geq t(j)} \frac{1}{d} z(j,t)$
Subject to:		Subject to:	
For each $t \in T$	$\sum_{j\ \mid\ t\geq t(j)}\sum_{t'\geq t} y(j,t') \leq 1$	For each $j, t \mid t \geq t(j)$:	$\sum_{k=t(j)}^{k=t} x_k + z(j,t) \geq 1$
For each $j, t \mid t \geq t(j)$:	$y(j,t) \leq \frac{1}{d}$		

Figure 6: The fractional TCP problem (the primal) and the corresponding dual problem

A Dynamic TCP Acknowledgement Problem

In this section we describe an algorithm for the dynamic TCP acknowledgement problem which is similar in spirit to the online algorithm we presented for the ski rental problem. The TCP acknowledgment problem was introduced by Dooly, Goldman and Scott [9] who gave a 2-competitive algorithm for the problem. This bound was later improved by [12] to a randomized $(1 - 1/e)$-competitive algorithm. We show an algorithm that is based on our primal dual approach, yielding the same competitive ratio as [12]. The dynamic TCP acknowledgment problem is the following. A stream of packets arrives at a destination and needs to be acknowledged. However, it is possible to acknowledge several packets by a single acknowledgement message. This can save on communication, but requires delaying the acknowledgement of certain messages (which is undesirable). Thus, the objective function is to minimize the number of acknowledgement messages sent along with the sum of latencies of the packets.

Let M be the set of packets. For each packet $j \in M$, let $t(j)$ be the time of arrival at the destination. Assume now that packets can only arrive in discrete times of $\frac{1}{d}$. We later take $d \to \infty$ so this assumption is not limiting. With the time discretization assumption, we can formulate the TCP acknowledgement problem as a covering linear program which appears in Figure 6. In this formulation we have a variable x_t for each discrete time t which is set to 1 if the algorithm sends an acknowledgement message at t. For each packet j and time $t \geq t(j)$, we have a variable $z(j,t)$ which is set to 1 if packet j is delayed between time t and time $t + \frac{1}{d}$. By this formulation, our objective is minimizing $\sum_{t\in T} x_t + \sum_{j\in M}\sum_{t\mid t\geq t(j)} \frac{1}{d} z(j,t)$. For each j and $\{t \mid t \geq t(j)\}$, we require that $\sum_{k=t(j)}^{t} x_k + z(j,t) \geq 1$. This guarantees that either the packet is delayed between time t and time $t + \frac{1}{d}$, or some acknowledgement message was sent since the arrival time of the packet. The dual packing problem contains variables $y(j,t)$ for each packet j and $t \geq t(j)$. We next design a primal-dual based algorithm for the problem.

Initially, $\forall k\ \ x_k \leftarrow 0$.
At each discrete time t (iteration), consider each of the packets j for which $\sum_{k=t(j)}^{k=t} x_k < 1$.
For each such packet j do the following update:

1. $z(j,t) \leftarrow 1 - \sum_{k=t(j)}^{k=t} x_k$
2. $x_t \leftarrow x_t + \frac{1}{d}\sum_{k=t(j)}^{k=t} x_k + \frac{1}{(c-1)\cdot d}$ (c is determined later).
3. $y(j,t) \leftarrow \frac{1}{d}$.

The analysis is not very difficult: First, the primal solution we produce is feasible. This follows since we update for each unsatisfied packet $z(j,t) \leftarrow 1 - \sum_{k=t(j)}^{k=t} x_k$ in each time t.

The second observation is that for each packet j and time t which we update, the change in the dual profit is $\frac{1}{d}$, while the change in our primal cost is:

$$\left(1 - \sum_{k=t(j)}^{k=t} x_k\right)\frac{1}{d} + \frac{1}{d}\left(\sum_{k=t(j)}^{k=t} x_k + \frac{1}{c-1}\right) = \frac{1}{d}\left(1 + \frac{1}{c-1}\right).$$

Dual (Packing)		Primal (Covering)	
Maximize:	$\sum_{i=1}^{n}\sum_{k=1}^{\ell} b(i,j,k)\,(1-x(i))\,y(i,j,k)$	Minimize :	$\sum_{i=1}^{n} s(i,j) + \sum_{k=1}^{\ell} z(j,k)$
Subject to:		Subject to:	
$\forall 1 \le k \le \ell$:	$\sum_{i=1}^{n} y(i,j,k) \le 1$	$\forall (i,k)$:	$s(i,j) + z(j,k) \ge b(i,j,k)\,((1-x(i))$
$\forall 1 \le i \le n$:	$\sum_{k=1}^{\ell} y(i,j,k) \le 1$	$\forall i,k$:	$s(i,j), z(j,k) \ge 0$
$\forall i,k$:	$y(i,j,k) \ge 0$		

Figure 7: The matching problem solved for product j. Here $x(i)$, $1 \le i \le n$, is a constant.

Finally, we want to choose c such that the dual solution we produce is feasible. Consider a time t and a corresponding dual constraint $\sum_{j \mid t \ge t(j)} \sum_{t' \ge t} y(j,t') \le 1$. We want that after d updates of $y(j,t')$ that "belong" to the constraint, all packets that have arrived prior to t are satisfied, and therefore there are no more updates of $y(j,t')$ belonging to the constraint. We prove that after d such updates, $\sum_{k \ge t} x_k \ge 1$, and so all packets that have arrived until time t are satisfied.

We prove by induction on the updates that $\sum_{k \ge t} x_k \ge \frac{(1+1/d)^q - 1}{c-1}$, where q is the number of updates. Before the first update, the claim trivially holds. Consider an update of $y(j,t')$ (at time t') such that $t' \ge t$ and packet j arrived at time $\le t$. By the algorithm we get that:

$$x_{t'} \leftarrow x_{t'} + \frac{1}{d}\sum_{k=t(j)}^{k=t'} x_k + \frac{1}{(c-1)\cdot d} \ge x_{t'} + \frac{1}{d}\sum_{k=t}^{k=t'} x_k + \frac{1}{(c-1)\cdot d}$$

Therefore $\sum_{k \ge t} x_k$ after the update is at least:

$$(1+1/d)\sum_{k \ge t} x_k + \frac{1}{(c-1)\cdot d} \ge (1+1/d)\,\frac{(1+1/d)^{q-1}-1}{c-1} + \frac{1}{(c-1)\cdot d} = \frac{(1+1/d)^q - 1}{c-1}$$

where the inequality follows by the induction hypothesis. Thus, choosing $c = (1+1/d)^d$ suffices, and thus when $d \to \infty$ we get a $(1-1/e)$ competitive algorithm. We note also that using the techniques in Section 5, it is possible to improve the competitive ratio to $C(d) = 1 - \frac{d-1}{d\left(1+\frac{1}{d-1}\right)^{d-1}}$. This is valid provided that packets only arrive in discrete times of $1/d$ units each.

In order to get a randomized integral solution we arrange the variables x_t on the infinite line. We choose a random number $p \in_R [0,1]$. We then send an acknowledgement message at each time segment x_t that falls in $p+k$ for some integer value k. We remark that we need the random choices to be correlated. It can be verified that our expected cost is the same as the cost of our fractional algorithm, completing the analysis.

B Proofs of Theorems

Proof of Theorem 3.2: We prove three simple claims:

1. The algorithm produces a primal feasible solution.
2. In each iteration, $\Delta P \le \left(1 + \frac{1}{c-1}\right) \cdot \Delta D$.
3. The algorithm produces an almost feasible dual solution.

To prove the claims, we crucially use the fact that the algorithm finds a maximum weight (integral) matching in H via a primal-dual algorithm. The primal and dual matching programs are in Figure 7. The algorithm outputs an optimal primal and dual solutions satisfying:

$$\sum_{i=1}^{n}\sum_{k=1}^{\ell} b(i,j,k)\,(1-x(i))\,y(i,j,k) = \sum_{i=1}^{n} s(i,j) + \sum_{k=1}^{\ell} z(j,k)$$

Proof of (1)**:** Recall that the primal constraint in the linear program of the multiple slot problem (see Figure 3) is:

$$\forall i,j,k: \quad b(i,j,k)x(i) + z(j,k) + s(i,j) \geq b(i,j,k).$$

Since $z(j,k) + s(i,j) \geq b(i,j,k)\,((1-x(i))$, the above constraint is satisfied.
Proof of (2)**:** When the jth product arrives,

$$\begin{aligned} \Delta P &= \sum_{i=1}^{n} z(j,i) + \sum_{k=1}^{\ell} s(j,i) + \sum_{i=1}^{n} B(i)\Delta x(i) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{\ell} b(i,j,k)\,(1-x(i))\,y(i,j,k) + \sum_{i=1}^{n}\sum_{k=1}^{\ell} B(i)\left(\frac{b(i,j,k)x(i)y(i,j,k)}{B(i)} + \frac{b(i,j,k)y(i,j,k)}{(c-1)\cdot B(i)}\right) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)\left(1+\frac{1}{c-1}\right) \end{aligned}$$

Since $\Delta D = \sum_{i=1}^{n}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)$, the claim follows.
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 $\sum_{j=1}^{m}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k) \geq B(i)$, then $x(i) \geq 1$. This is done by showing that

$$x(i) \geq \frac{1}{c-1}\left(c^{\frac{\sum_{j=1}^{m}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)}{B(i)}} - 1\right). \tag{5}$$

Thus, whenever $\sum_{j=1}^{m}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k) \geq B(i)$, we get that $x(i) \geq 1$. We prove (5) by induction on the (relevant) iterations of the algorithm. Initially, this assumption is trivially true. We are only concerned about iterations in which the kth slot of product t is sold to buyer i. In such an iteration we get that:

$$\begin{aligned} x(i)_{\text{end}} &= x(i)_{\text{start}}\cdot\left(1+\frac{b(i,t,k)}{B(i)}\right) + \frac{b(i,t,k)}{(c-1)\cdot B(i)} \\ &\geq \frac{1}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{t\}}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)}{B(i)}} - 1\right]\cdot\left(1+\frac{b(i,t,k)}{B(i)}\right) + \frac{b(i,t,k)}{(c-1)\cdot B(i)} \qquad (6) \\ &= \frac{1}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{t\}}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)}{B(i)}}\cdot\left(1+\frac{b(i,t,k)}{B(i)}\right) - 1\right] \\ &\geq \frac{1}{c-1}\left[c^{\frac{\sum_{j\in J\setminus\{t\}}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)}{B(i)}}\cdot c^{\left(\frac{b(i,t,k)}{B(i)}\right)} - 1\right] = \frac{1}{c-1}\left[c^{\frac{\sum_{j\in M}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k)}{B(i)}} - 1\right]. \qquad (7) \end{aligned}$$

Where Inequality (6) follows from the induction hypothesis, and Inequality (7) follows since, for any $0 \leq x \leq y \leq 1$, $\frac{\ln(1+x)}{x} \geq \frac{\ln(1+y)}{y}$.

By the above, it follows that whenever the sum of the charges to a buyer is more than its budget, we stop charging this buyer. Thus, there can be at most one iteration in which we charge the buyer by less than

$b(i,j,k)$. Therefore, for each buyer i: $\sum_{j\in M}\sum_{k=1}^{\ell} b(i,j,k)y(i,j,k) \le B(i) + \max_{j\in M,k}\{b(i,j,k)\}$, and thus the profit extracted from buyer i is at least:

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

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

Proof of Theorem 4.1: We first prove a more general claim regarding the final value of $x(i)$. During the algorithm we increase the value of primal variables $x(i)$. For buyer i, let $x(i,\text{end})$ be the final (highest) value of $x(i)$ (upon termination). By our assumption, buyer i extracted at least g_i fraction of its budget. Whenever we charge a buyer i for an item and $x(i) < x_s(i)$, the algorithm updates:

$$x(i) \leftarrow x(i) + \frac{b(i,j)}{B(i)}\left(x_s(i) + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\right)$$

Thus, the final value of $x(i)$ is:

$$x(i,\text{end}) \ge g_i \cdot \left(x_s(i) + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\right) = g_i \cdot x_s(i) + (1-g_i)x_s(i) = x_s(i) \tag{8}$$

We next prove three simple claims:

- The algorithm produces a primal feasible solution.
- In each iteration, $\Delta P \le (1 + \frac{1-g}{c^{1-g}-(1-g)}) \cdot \Delta D$.
- The algorithm produces an almost feasible dual solution.

Proof of (1)**:** Consider a primal constraint of buyer i and any item j. In order to make this constraint feasible, we need to set $z(j) \ge \max\{0, b(i,j)(1-x(i,\text{end}))\}$. By Equation 8, $x(i,\text{end}) \ge x_s(i)$. Thus, when item j arrives, setting $z(j)$ to be $b(i,j)(1-\max\{x(i),x_s(i)\}) \ge b(i,j)(1-x(i,\text{end}))$ suffices to satisfy the constraint. Since the algorithm chooses the buyer i that maximizes this value, and sets $z(j)$ according to this maximal value, we get that the constraint corresponding to any buyer i and item j is satisfied.
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:

$$\begin{aligned} B(i)\Delta x(i) + z(j) &= B(i)\cdot\left(\frac{b(i,j)\max\{x(i),x_s(i)\}}{B(i)} + \frac{b(i,j)}{B(i)}\frac{1-g_i}{c^{1-g_i}-(1-g_i)}\right) \\ &+ b(i,j)(1-\max\{x(i),x_s(i)\}) \\ &= b(i,j)\left(1+\frac{1-g_i}{c^{1-g_i}-(1-g_i)}\right) \le b(i,j)\left(1+\frac{1-g}{c^{1-g}-(1-g)}\right) \end{aligned}$$

Proof of (3)**:** The algorithm never updates the dual solution for buyers satisfying $x(i) \ge 1$. We prove that for any buyer i, when $\sum_{j\in M} b(i,j)y(i,j) \ge B(i)$, then $x(i) \ge 1$. This is done by proving that if the buyer i extracted g'_i fraction of its budget (i.e. $\sum_{j\in M} b(i,j)y(i,j) = g'_i \cdot B(i)$) then:

$$x(i) \ge \begin{cases} g'_i\left[x_s(i) + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\right] & \text{if } g'_i \le g_i \\ x_s(i)c^{g'_i-g_i} + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\left[c^{g'_i-g_i}-1\right] & \text{if } g'_i > g_i \end{cases} \tag{9}$$

It is easy to check that when $g'_i = g_i$, the two are the same and equal to $x_s(i)$. Thus, if the claim is correct, then whenever buyer i extracts all his budget we get that:

$$\begin{aligned} x(i) &\geq x_s(i)c^{1-g_i} + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\left[c^{1-g_i}-1\right] \\ &= \frac{g_i}{c^{1-g_i}-(1-g_i)}c^{1-g_i} + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\left[c^{1-g_i}-1\right] = 1 \end{aligned}$$

We prove Inequality 9 by induction on the (relevant) iterations of the algorithm. Initially, this assumption is trivially true. We are only concerned about iterations in which an item, say k, is sold to buyer i. Let g'_i be the fraction of the budget buyer i spent before the current allocation, and let $g''_i = g'_i + \frac{b(i,j)}{B(i)}$ be the fraction of the budget buyer i spends after the current allocation. In iterations in which $x(i) < x_s(i)$, we get by Equality 8 that $g'_i < g_i$, and thus:

$$\begin{aligned} x(i)_{\text{end}} &= x(i)_{\text{start}} + x_s(i)\frac{b(i,k)}{B(i)} + \frac{b(i,k)}{B(i)}\frac{1-g_i}{c^{1-g_i}-(1-g_i)} \\ &\geq g'_i\left[x_s(i) + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\right] + x_s(i)\frac{b(i,k)}{B(i)} + \frac{b(i,k)}{B(i)}\frac{1-g_i}{c^{1-g_i}-(1-g_i)} \qquad (10) \\ &= g''_i\left[x_s(i) + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\right] \end{aligned}$$

Where Inequality 10 follows by the induction hypothesis. We also remark here that if the budget extracted from buyer i before the iteration is less than g_i, and the budget extracted after the iteration is strictly more than g_i, then it is possible to divide the cost of the item $b(i,j)$ into two costs $b(i,j)_1 + b(i,j)_2 = b(i,j)$, such that the budget extracted after virtually selling $b(i,j)_1$ is exactly g_i. We virtually sell both items to buyer i and change $x(i)$ in two iterations. It is easy to verify that the change of $x(i)$ is the same as if this was done in a single iteration.

In iterations in which $x(i) \geq x_s(i)$ we get by Equality 8 that $g'_i \geq g_i$ and so:

$$\begin{aligned} x(i)_{\text{end}} &= x(i)_{\text{start}}\left(1 + \frac{b(i,k)}{B(i)}\right) + \frac{b(i,k)}{B(i)}\frac{1-g_i}{c^{1-g_i}-(1-g_i)} \\ &\geq \left[x_s(i)c^{g'_i-g_i} + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\left[c^{g'_i-g_i}-1\right]\right]\left(1 + \frac{b(i,k)}{B(i)}\right) + \frac{b(i,k)}{B(i)}\frac{1-g_i}{c^{1-g_i}-(1-g_i)} \qquad (11) \\ &= x_s(i)c^{g'_i-g_i}\left(1 + \frac{b(i,k)}{B(i)}\right) + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\left(c^{g'_i-g_i}\left(1 + \frac{b(i,k)}{B(i)}\right) - 1\right) \\ &\geq x_s(i)c^{g'_i-g_i}\cdot c^{\left(\frac{b(i,k)}{B(i)}\right)} + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\left(c^{g'_i-g_i}\cdot c^{\left(\frac{b(i,k)}{B(i)}\right)} - 1\right) \qquad (12) \\ &= x_s(i)c^{g''_i-g_i} + \frac{1-g_i}{c^{1-g_i}-(1-g_i)}\left[c^{g''_i-g_i}-1\right] \end{aligned}$$

Where Inequality 11 follows by the induction hypothesis, and Inequality 12 follows since for any $0 \leq x \leq y \leq 1$, $\frac{\ln(1+x)}{x} \geq \frac{\ln(1+y)}{y}$.

By the above, it follows that whenever the sum of the charges to a buyer is more than its budget, we stop charging this buyer. Thus, there can be at most one iteration in which we charge the buyer by less than $b(i,j)$. Therefore, for each buyer i: $\sum_{j\in M} b(i,j)y(i,j) \leq B(i) + \max_{j\in M}\{b(i,j)\}$, and thus the profit extracted from buyer i is at least:

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

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

Proof of Theorem 5.1: Let $Y(j)$ denote the total profit of the algorithm (the dual packing) in the jth iteration. In each iteration we maintain a corresponding feasible primal solution whose value is denoted by $X(j)$. Upon arrival of a new product we update both primal and dual programs. The dual (packing) program is updated by adding a new constraint corresponding to the new product which has arrived, and by adding a new term $b(j)y(i,j)$ to each constraint of an interested buyer. The primal program is updated by adding a new variable $z(j)$ for the new product and a constraint of the form $b(j)x(i)+z(j)\geq b(j)$ for each buyer who is interested in the new product.

Initially, the dual and primal programs are empty. In the jth iteration, the change in values of the primal and dual solutions is denoted by $\Delta X(j)$ and $\Delta Y(j)$, correspondingly. We prove that in each iteration:

$$\Delta X(j) \leq \frac{1}{C(d)}\cdot \Delta Y(j)$$

The primal solution is an assignment of values to the variables $x(i)$ and $z(j)$. Since these values are not used by the allocation algorithm, we can set them using future knowledge. For each buyer i, let $t(i)$ ($0\leq t\leq d$) be the largest level i to which this buyer belongs during the algorithm. Thus, buyer i spent overall at least $t(i)/d$ fraction of his budget. The variable $x(i)$ grows as a function of the fraction of money that buyer i spent, which in fact depends on the corresponding dual constraint. Specifically, for buyer i:

$$x(i)=\begin{cases} f_d\left(\frac{1}{B(i)}\sum_{j\,\mid\, i\in S(j)} b(j)y(i,j)\right) & \text{if } \frac{1}{B(i)}\sum_{j\,\mid\, i\in S(j)} b(j)y(i,j)\leq \frac{t(i)}{d} \\ f_d\left(\frac{t(i)}{d}\right) & \text{if } \frac{1}{B(i)}\sum_{j\,\mid\, i\in S(j)} b(j)y(i,j)\geq \frac{t(i)}{d} \end{cases}$$

The variables $x(i)$ are monotonically increasing and thus, once a primal constraint is satisfied, it remains satisfied throughout the run of the algorithm. Hence, in each iteration, it suffices to satisfy the newly added primal constraints.

Consider first a case in which product j was not fully sold by the algorithm. This means that at the end of the jth iteration all the buyers in $S(j)$ exhausted their budget. In this case the corresponding variables $x(i)$ at the end of the iteration are all 1, and thus all the new primal constraints are satisfied, and we can set $z(j)\leftarrow 0$. We only need to show that the change in the primal profit in this iteration is not too large. When we increase a variable $y(i,j)$, the derivative of the dual profit of the algorithm is $b(j)$. The derivative of the primal cost is:

$$B(i)\cdot\frac{df_d}{d(y(i,j))}\leq B(i)\cdot\frac{b(j)}{B(i)}\cdot d\cdot a_d=\frac{1}{C(d)}\cdot b(j).$$

The inequality follows by taking the maximum derivative of the (convex) function f_d which is:

$$d\cdot a_d = da_1\left(1+\frac{1}{d-1}\right)^{d-1}=\frac{1}{C(d)}.$$

Thus, we get that in this iteration $\Delta X(j)\leq\frac{1}{C(d)}\cdot\Delta Y(j)$.

Assume now that product j was fully sold to the buyers. Let t, $0\leq t\leq d-1$, be the highest level of buyers to which the product was sold. Since the algorithm always allocates the product to buyers in the lowest possible level it means that all buyers in $S(j)$ used at least t/d fraction of their money. Let $\Delta_0, \Delta_1, \ldots, \Delta_t$ be the fraction of the product that was allocated in each level $k \leq t$. By our assumption: $\sum_{k=1}^{t} \Delta_k = 1$. We consider two cases.

Case 1**:** All the buyers in $S(j)$ spend during the algorithm at least t'/d of their budget for $t' > t$. In this case, for each buyer i, the derivative of the primal cost due to the change in $x(i)$ is:

$$B(i) \cdot \frac{df_d}{d(y(i,j))} \leq B(i) \cdot \frac{b(j)}{B(i)} \cdot d \cdot a_{t+1} = b(j) \cdot d \cdot a_{t+1}.$$

The inequality follows by taking the derivative of f_d in the highest level in which the product was sold. We fully allocate the product and hence $\sum_{i \in S(j)} y(i,j) = 1$. Thus, the total change of the primal cost due to the change in the variables $x(i)$ is at most $b(j) \cdot d \cdot a_{t+1}$. Since all buyers in $S(j)$ eventually spend during the algorithm at least $(t+1)/d$ of their budget, variable $x(i)$ corresponding to buyer $i \in S(j)$ will be at the end of the allocation process at least $f(\frac{t+1}{d})$. Therefore, it is safe to set $z(j) = b(j) \cdot (1 - f(\frac{t+1}{d}))$ in order to satisfy all the new primal constraints. Thus, the total change in the primal cost in this iteration is:

$$\begin{aligned}
z(j) + \sum_{i \in S'(j)} B(i)\Delta(x(i)) &\leq b(j)\left(1 - f(\frac{t+1}{d})\right) + b(j) \cdot d \cdot a_{t+1} \\
&= b(j)\left(1 - a_1 \cdot \left[d\left(1 + \frac{1}{d-1}\right)^t - (d-1)\right]\right) + b(j) \cdot d \cdot a_1 \cdot \left(1 + \frac{1}{d-1}\right)^t \\
&= b(j)\left(1 + a_1 \cdot (d-1)\right) = b(j)\left(1 + \frac{d-1}{d\left(1+\frac{1}{d-1}\right)^{d-1} - (d-1)}\right) = \frac{1}{C(d)} \cdot b(j).
\end{aligned}$$

Since the product was fully sold the dual profit in this case is $b(j)$ and hence we are done with this case.

Case 2**:** There exists at least one buyer in $S(j)$ who eventually spends (throughout the algorithm) less than a fraction of $(t+1)/d$ of his budget (but spend at least t/d). In this case, in order to satisfy the new primal constraint, it is only safe to set $z(j) = b(j) \cdot (1 - f(\frac{t}{d}))$. However, note that the buyer that spent less than $(t+1)/d$ fraction of its money was present throughout the whole process of dividing the product equally between all buyers in last level t. Thus, by our algorithm, this buyer receives at least a fraction $\frac{\Delta_t}{d}$ of the product. By the definition of the function associated with the variable $x(i)$, the growth function of $x(i)$ in this segment (which is larger than $t(i)$) is zero. Thus, the change in the primal cost due to the increase of the dual variables in the highest level is at most:

$$b(j)a_{t+1}d \cdot \frac{d-1}{d}\Delta_t = b(j) \cdot (d-1) \cdot a_{t+1} \cdot \Delta_t. \tag{13}$$

The change in the primal cost due to the increase of the dual variables in lower levels is at most $b(j) \cdot d \cdot a_t \cdot (1 - \Delta_t)$. But, $a_{t+1} = a_t \cdot \left(1 + \frac{1}{d-1}\right)$, and so $a_t = \frac{d-1}{d} \cdot a_{t+1}$. Thus, the change in the primal cost due to change in the variables $x(i)$ is at most:

$$b(j) \cdot d \cdot a_t \cdot (1 - \Delta_t) = b(j) \cdot d \cdot \frac{d-1}{d} \cdot a_{t+1} \cdot (1 - \Delta_t) = b(j) \cdot (d-1) \cdot a_{t+1} \cdot (1 - \Delta_t). \tag{14}$$

Adding up Equations (13) and (14), we get that the total change in the primal cost due to the increase in the primal variables $x(i)$ is $b(j)(d-1)a_{t+1}$. Since $f(\frac{t+1}{d}) = a_{t+1} + f(\frac{t}{d})$, the total change in the primal cost is at most:

$$\begin{aligned}
b(j)\left(1 - f(\frac{t}{d})\right) + b(j) \cdot (d-1) \cdot a_{t+1} &= b(j)\left(1 - f(\frac{t+1}{d}) + a_{t+1})\right) + b(j) \cdot (d-1) \cdot a_{t+1} \\
&= b(j)\left(1 - f(\frac{t+1}{d})\right) + b(j)da_{t+1} = \frac{1}{C(d)} \cdot b(j).
\end{aligned}$$

This change is exactly the same as in case (1). Similarly to case (1), the product was fully sold and so the dual profit is $b(j)$ and we are done with this case.

Proof of Lemma 5.3: Initially, $\Phi = 1$, and it can easily be verified that subsequently Φ is always positive.

The proof of the second claim is probabilistic. Assume that in iteration j the product is allocated fractionally. We choose to virtually allocate the product to buyer i with probability $y(i,j)$. This can be done by simply arranging the values $y(i,j)$ on the interval $[0,1]$ and choosing uniformly in random a number in $[0,1]$. The probability of virtually allocating product j to buyer i is then exactly $y(i,j)$, and the probability of not allocating (virtually) the product to any of the buyers is $1 - \sum_{i\in S(j)} y(i,j)$.

We prove that the expected value of the potential function Φ does not increase. Thus, either there exists a buyer such that virtually allocating product j to it does not increase Φ, or by not allocating product j at all Φ does not increase. By linearity of expectation, we can analyze each of the terms in Φ corresponding to buyers in $S(j)$ separately. Let $\Phi^S_{1,i}$ and $\Phi^E_{1,i}$ be the values of the term corresponding to the ith buyer in the first term of the potential function, Φ_1, before and after the probabilistic choice, correspondingly. Thus,

$$\begin{aligned}
E\left[\Phi^E_{1,i}\right] &= y(i,j)\cdot\Phi^S_{1,i}\exp\left\{\ln\left(1+\sqrt{\frac{\ln 2n}{B(i)}}\cdot b(j)\right)\cdot 1-\sqrt{\frac{\ln 2n}{B(i)}}\cdot b(j)y(i,j)\right\}\\
&+ (1-y(i,j))\cdot\Phi^S_{1,i}\exp\left\{-\sqrt{\frac{\ln 2n}{B(i)}}\cdot b(j)y(i,j)\right\}\\
&= \Phi^S_{1,i}\left(\left(1+\sqrt{\frac{\ln 2n}{B(i)}}\cdot b(j)y(i,j)\right)\exp\left\{-\sqrt{\frac{\ln 2n}{B(i)}}\cdot b(j)y(i,j)\right\}\right)\le\Phi^S_{1,i},
\end{aligned}$$

where the last inequality follows since $1+x\le e^x$ for $x\ge 0$.

Next, we analyze the second term of the potential function, Φ_2. Let Φ^S_2 and Φ^E_2 be the values of this term before and after the probabilistic choice, correspondingly. Let $y(j) = \sum_{i\in S(j)} y(i,j)$. $y(j)$ is exactly the probability that product j is allocated by the probabilistic choice. It can be verified that the value of Φ_2 only depends on the fact that the product is virtually allocated or not, and does not depend on the identity of the buyer to whom the product was allocated. Thus,

$$\begin{aligned}
E\left[\Phi^E_2\right] &= y(j)\cdot\Phi^S_2\exp\left\{\frac{1}{\sqrt{B_{\min}}}b(j)y(j)+\ln\left(1-\frac{b(j)}{\sqrt{B_{\min}}}\right)\right\}+(1-y(j))\cdot\Phi^S_2\exp\left\{\frac{1}{\sqrt{B_{\min}}}b(j)y(j)\right\}\\
&= y(j)\left(1-\frac{b(j)}{\sqrt{B_{\min}}}\right)\cdot\Phi^S_2\exp\left\{\frac{1}{\sqrt{B_{\min}}}b(j)y(j)\right\}+(1-y(j))\cdot\Phi^S_2\exp\left\{\frac{1}{\sqrt{B_{\min}}}b(j)y(j)\right\}\\
&= \Phi^S_2\exp\left\{\frac{1}{\sqrt{B_{\min}}}b(j)y(j)\right\}\left(1-y(j)\frac{b(j)}{\sqrt{B_{\min}}}\right)\le\Phi^S_2,
\end{aligned}$$

where the last inequality follows since $1-x\le e^{-x}$ for $x\ge 0$.

Proof of Theorem 5.4: Consider first Φ_2. Since each term in Φ is positive, and $\Phi\le 1$, then also $\Phi_2\le 1$. Thus,

$$\frac{1}{2}\exp\left\{\sum_{i=1}^{n}\sum_{j\mid i\in S(j)}\frac{1}{\sqrt{B_{\min}}}b(j)y(i,j)+\sum_{i=1}^{n}\sum_{j\mid i\in S(j)}\ln\left(1-\frac{b(j)}{\sqrt{B_{\min}}}\right)\chi(i,j)\right\}\le 1.$$

Simplifying the inequality we get that:

$$\sum_{i=1}^{n}\sum_{j\mid i\in S(j)} b(j)y(i,j) \quad\le\quad \sqrt{B_{\min}}\ln 2+\sqrt{B_{\min}}\sum_{i\in M}\sum_{j\mid i\in S(j)}\ln\left(1-\frac{b(j)}{\sqrt{B_{\min}}}\right)^{-1}\chi(i,j)$$

$$\leq \sqrt{B_{\min}} \ln 2 + \sqrt{B_{\min}} \cdot \sum_{i \in M} \sum_{j \mid i \in S(j)} \frac{\frac{b(j)}{\sqrt{B_{\min}}}}{1 - \frac{b(j)}{\sqrt{B_{\min}}}} \chi(i,j) \tag{15}$$

$$= \sqrt{B_{\min}} \ln 2 + \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} \frac{1}{1 - \frac{b(j)}{\sqrt{B_{\min}}}} b(j) \chi(i,j).$$

Inequality (15) follows since for any $0 < x \leq 1$, $\ln\left(\frac{1}{1-x}\right) = \ln\left(1 + \frac{x}{1-x}\right) \leq \frac{x}{1-x}$.

If $\sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)y(i,j) \leq B_{min}$, then each product that is virtually allocated is also allocated and, $\sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)\chi(i,j) \geq \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)y(i,j)$. Otherwise, (simplifying the previous inequality) we get that:

$$\sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)\chi(i,j) \geq \left(1 - \frac{b_{\max}}{\sqrt{B_{\min}}} - \frac{\ln 2}{\sqrt{B_{\min}}}\right) \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)y(i,j)$$

$$= (1 - o(1)) \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)y(i,j) \tag{16}$$

Thus, in both cases, the total revenue of a product that the integral algorithm virtually allocates is at least $1 - o(1)$ times the total revenue of the fractional solution. We now need to analyze the loss in revenue of the algorithm in the case where a product is virtually allocated to a buyer, but due to insufficient budget, the product is not allocated to the buyer. We use the first part of the potential function to prove that buyers do not exceed their budget in the integral solution by much compared with the budget used in the fractional allocation. Since each term in Φ is positive, and $\Phi \leq 1$, then each term Φ can be at most 1. Consider a term corresponding to a buyer i. We get that:

$$\frac{1}{2n} \exp\left\{ \sum_{j \mid i \in S(j)} \ln\left(1 + \sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j)\right) \chi(i,j) - \sum_{j \mid i \in S(j)} \sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j)y(i,j) \right\} \leq 1.$$

Simplifying we get:

$$\sum_{j \mid i \in S(j)} b(j)y(i,j) \geq -\sqrt{B(i) \ln 2n} + \sum_{j \mid i \in S(j)} \frac{\ln\left(1 + \sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j)\right)}{\sqrt{\frac{\ln 2n}{B(i)}} b(j)} b(j)\chi(i,j).$$

Consider now a buyer for which $\sum_{j \mid i \in S(j)} b(j)\chi(i,j) \geq B(i)$. In this case we get that:

$$\sum_{j \mid i \in S(j)} b(j)y(i,j) \geq -\sqrt{B(i) \ln 2n} + \sum_{j \mid i \in S(j)} \frac{\ln\left(1 + \sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j)\right)}{\sqrt{\frac{\ln 2n}{B(i)}} b(j)} b(j)\chi(i,j)$$

$$\geq \sum_{j \mid i \in S(j)} \left[\frac{\ln\left(1 + \sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j)\right)}{\sqrt{\frac{\ln 2n}{B(i)}} b(j)} - \sqrt{\frac{\ln 2n}{B(i)}} \right] b(j)\chi(i,j) \tag{17}$$

$$\geq \sum_{j \mid i \in S(j)} \left[\frac{\sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j) - \frac{1}{2}\left(\sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j)\right)^2}{\sqrt{\frac{\ln 2n}{B(i)}} b(j)} - \sqrt{\frac{\ln 2n}{B(i)}} \right] b(j)\chi(i,j) \tag{18}$$

$$= \sum_{j \mid i \in S(j)} \left[1 - \frac{1}{2}\sqrt{\frac{\ln 2n}{B(i)}} \cdot b(j) - \sqrt{\frac{\ln 2n}{B(i)}} \right] b(j)\chi(i,j) = \sum_{j \mid i \in S(j)} [1 - o(1)]\, b(j)\chi(i,j),$$

Inequality (17) follows since for these buyers $\sum_{j \mid i \in S(j)} b(j)\chi(i,j) \geq B(i)$. Inequality (18) follows since for any $x \geq 0$, $\ln(1+x) \geq x - \frac{1}{2}x^2$. The last equality follows from (4).

We now partition the buyers into two sets, M_1 and M_2, where the set M_1 contains buyers for which $\sum_{j \mid i \in S(j)} b(j)\chi(i,j) \geq B(i)$, and the set M_2 contains the remaining buyers. The total revenue of the algorithm is:

$$
\begin{aligned}
P &= \sum_{i \in M_1} B(i) + \sum_{i \in M_2} \sum_{j \mid i \in S(j)} b(j)\chi(i,j) \\
&\geq \sum_{i \in M_1} \sum_{j \mid i \in S(j)} b(j)y(i,j) + \sum_{i \in M_2} \sum_{j \mid i \in S(j)} b(j)\chi(i,j) && (19) \\
&= (1 - o(1)) \sum_{i \in M_1} \sum_{j \mid i \in S(j)} b(j)\chi(i,j) + \sum_{i \in M_2} \sum_{j \mid i \in S(j)} b(j)\chi(i,j) && (20) \\
&\geq (1 - o(1)) \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)\chi(i,j) \geq (1 - o(1)) \sum_{i=1}^{n} \sum_{j \mid i \in S(j)} b(j)y(i,j). && (21)
\end{aligned}
$$

Inequality (19) follows since the fractional solution is feasible, and thus, for the buyers in M_1 it holds that $\sum_{j \mid i \in S(j)} b(j)\chi(i,j) \geq B(i) \geq \sum_{j \mid i \in S(j)} b(j)y(i,j)$. Inequality (20) follows from our above observation and inequality (21) follows from Inequality (16). Thus, we get that the loss of revenue of the integral solution is bounded by a factor of $o(1)$ with respect to the fractional solution, thus completing the proof.

Proof of Lemma 5.5: We prove that the bound holds even for an algorithm that is allowed to allocate the products fractionally. We reconsider the standard lower bound example. For any d, the instance consists of d buyers, each has unit budget 1. Each of the products has unit price. The adversarial arrival sequence is as follows. There are d products. The first product can be allocated to all d buyers. The next product cannot be allocated to the buyer to whom the online algorithm allocated the smallest fraction of the first product. The third product can be allocated only to the $d-2$ buyers who got most of the previous two items. This sequence continues until the last product, which can only be allocated to a single buyer.

It is not hard to see that the optimal (maximum) revenue is d. Any deterministic algorithm allocates to the "poorest" buyer an amount of at most $\frac{1}{d}$. The second (poorest) buyer gets a fraction of $\frac{1}{d} + \frac{1}{d-1}$. In general, the revenue extracted from the ith poorest buyer is $\min\{1, H(d) - H(d-i)\}$. Thus, the competitive ratio of any algorithm is at most:

$$
\begin{aligned}
\frac{1}{d}\sum_{i=1}^{d} \min\{1, H(d) - H(d-i)\} &= \frac{1}{d}\left(1 \cdot (d-k) + k \cdot H(d) - \sum_{i=1}^{k} H(d-i)\right) \\
&= 1 - \frac{k - kH(d) + \sum_{i=1}^{k} H(d-i)}{d},
\end{aligned}
$$

where $H(\cdot)$ is the harmonic number, and k is the largest value for which $H(d) - H(d-k) \leq 1$

Proof of Lemma 5.6: We prove that the bound holds even for an algorithm that is allowed to allocate the products fractionally. We consider an instance with 9 buyers, each with unit budget. There are 9 products and has unit price. The adversarial sequence is as follows. The first three items can be bought by buyers $\{1,2,7\}, \{3,4,8\}, \{5,6,9\}$ respectively. Assume without loss of generality that buyers $7, 8, 9$ got at most $1/3$ of a product. No other products will be available for these buyers, so we can assume that the algorithm allocates exactly $1/3$ of a product to each of these buyers. Next, there are two products that can be bought by $\{1,2,5\}, \{3,4,6\}$, respectively. The total revenue out of these buyers is 4 and so we can assume that buyers 5 and 6 got at most $2/3$ of a product each. Next, the four last products arrive and they can each go only to buyers $1, 2, 3$ and 4, respectively.

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

Figure 8: The fractional generalized risk management ad-auction problem (the dual) and the corresponding primal problem

It is not hard to see that the optimal (maximum) revenue is 9. By the adversarial sequence any deterministic algorithm has revenue of at most: $4+2\cdot\frac{2}{3}+3\cdot\frac{1}{3}=\frac{19}{3}$. Thus, $C(d)\le\frac{19}{27}=0.703$.

Modified algorithm for risk management and the proof of Theorem 6.1: As states, the more general risk management problem can be formulated using a more complex linear program. The primal-dual pair is described in Figure 8. Our 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:

$$\max\left\{\max_{k=1}^{r_i-1}\{a(i,k)b(i,j)(1-x(i,k))\}, a(i,r_i)b(i,j)\right\}$$

Let $1\le r\le r_i$ be the the argument for which the maximization is achieved. If $1\le r\le r_i-1$ and $x(i,r)\ge 1$ then do nothing. Otherwise:

1. Charge the the budget $B(i,r)$ by the minimum between $a(i,r)b(i,j)$ and the remaining of the budget and set $y(i,j,r)\leftarrow 1$
2. If $r\ne r_i$ then:
   - $z(j)\leftarrow a(i,r)b(i,j)(1-x(i,r))$.
   - $x(i,r)\leftarrow x(i,r)(1+\frac{a(i,r)b(i,j)}{B(i,r)})+\frac{a(i,r)b(i,j)}{B(i,r)}\frac{1-a(i,r_i)}{c-1}$ (c is determined later).
3. If $r=r_i$ then $z(j)\leftarrow a(i,r_i)b(i,j)$.

Let $R_{\max}=\max_{i\in I,j\in M,1\le r\le r_i-1}\{\frac{a(i,r)b(i,j)}{B(i,r)}\}$, the maximum ratio between a charge to a budget and the total budget.

Theorem B.1. *(Theorem 6.1) The algorithm is* $\left(\frac{c-1}{c-a_{\min}}\right)(1-R_{\max})$*-competitive, where* $c=(1+R_{\max})^{\frac{1}{R_{\max}}}$*. When* $R_{\max}\to 0$ *the competitive ratio of the algorithm tends to* $\frac{e-1}{e-a_{\min}}$.

Proof. We prove three simple claims:

- The algorithm produces a primal feasible solution.
- In each iteration the $\Delta P\le(1+\frac{1-a_{\min}}{c-1})\cdot\Delta D$.

- The algorithm produces an almost feasible dual solution.

Proof of (1)**:** Consider some buyer i' and budget $B(i',k')$. To make this constraint feasible we should choose $z(j) \geq a(i',k')b(i',j)(1-x(i',k')$. Also for each buyer i and product j, in order to satisfy the last constraint we need $x(j) \geq a(i,r_i)b(i,j)$. Since the algorithm set $z(j)$ according to the maximal value all constraint become feasible. If the algorithm do not update $z(j)$, it means that all these values are at most 0 and so all the constraints are feasible.
Proof of (2)**:** Whenever the algorithm updates the primal and dual solutions, the change in the dual profit is $a(i,r)b(i,j)$. (Note that even if the remaining of budget $B(i,r)$ to which product j is being charged is less than $a(i,r)b(i,j)$, variable $y(i,j,r)$ is still set to 1.) If $r \neq r_i$ then the change in the primal cost is:

$$\begin{aligned} B(i,r)\Delta x(i,r) + z(j) &= B(i,r)\cdot\left(\frac{a(i,r)b(i,j)x(i,r)}{B(i,r)} + a(i,r)b(i,j)\frac{1-a(i,r_i)}{(c-1)\cdot B(i,k)}\right) \\ &+ a(i,r)b(i,j)(1-x(i,r)) \\ &\leq a(i,r)b(i,j)(1+\frac{1-a_{\min}}{c-1}) \end{aligned}$$

If $r = r_i$ then the change in the primal cost is: $z(j) = a(i,r)b(i,j)$.
Proof of (3)**:** The algorithm never charges budget $B(i,r)$ when $x(i,r) \geq 1-a(i,r_i)$. This follows since at that point:

$$a(i,r_i)b(i,j) \geq a(i,r)b(i,j)(1-(1-a(i,r_i))) \geq a(i,r)b(i,j)(1-x(i,r))$$

We prove that for any buyer i and $1 \leq r \leq r_i - 1$ when $\sum_{j\in M} a(i,r)b(i,j)y(i,j,r) \geq B(i,r)$ then $x(i,r) \geq\geq 1-a(i,r_i)$. This is done by proving that:

$$x(i,r) \geq \frac{1-a(i,r_i)}{c-1}\left[c^{\frac{\sum_{j\in M} a(i,r)b(i,j)y(i,j,r)}{B(i,r)}} - 1\right]. \tag{22}$$

Thus, whenever $\sum_{j\in M} a(i,r)b(i,j)y(i,j,r) \geq B(i,r)$, we get that $x(i,r) \geq 1-a(r_i)$. We prove (22) by induction on the (relevant) iterations of the algorithm. Initially, this assumption is trivially true. We only concern about iterations in which an item, say k, was sold to buyer i on the budget $B(i,r)$. In such an iteration we get that:

$$\begin{aligned} x(i,r)_{\text{end}} &= x(i,r)_{\text{start}}\cdot\left(1+\frac{a(i,r)b(i,k)}{B(i,r)}\right) + a(i,r)b(i,k)\frac{1-a(i,r_i)}{(c-1)\cdot B(i,r)} \\ &\geq \frac{1-a(i,r_i)}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{k\}} a(i,r)b(i,j)y(i,j,r)}{B(i,r)}} - 1\right]\cdot\left(1+\frac{a(i,r)b(i,k)}{B(i,r)}\right) \\ &+ a(i,r)b(i,k)\frac{1-a(i,r_i)}{(c-1)\cdot B(i,r)} \qquad (23) \\ &= \frac{1-a(i,r_i)}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{k\}} a(i,r)b(i,j)y(i,j,r)}{B(i,r)}}\cdot\left(1+\frac{a(i,r)b(i,k)}{B(i,r)}\right) - 1\right] \\ &\geq \frac{1-a(i,r_i)}{c-1}\left[c^{\frac{\sum_{j\in M\setminus\{k\}} a(i,r)b(i,j)y(i,j,r)}{B(i,r)}}\cdot c^{\left(\frac{a(i,r)b(i,k)}{B(i,r)}\right)} - 1\right] \qquad (24) \\ &= \frac{1-a(i,r_i)}{c-1}\left[c^{\frac{\sum_{j\in M} a(i,r)b(i,j)y(i,j,r)}{B(i,r)}} - 1\right] \end{aligned}$$

Where Inequality 23 follows by the induction hypothesis, and Inequality 24 follows since for any $0 \leq x \leq y \leq 1$, $\frac{\ln(1+x)}{x} \geq \frac{\ln(1+y)}{y}$.

By the above, it follows that whenever the sum of the charges of a buyer exceeds its budget, we stop charging the buyer. Thus, there can be at most one iteration in which we charge a budget $B(i,r)$ by less than $a(i,r)b(i,j)$. Therefore, for each budget $B(i,r)$: $\sum_{j\in M} a(i,r)b(i,j)y(i,j,r) \leq B(i,r)+\max_{j\in M}\{a(i,r)b(i,j)\}$, and thus the profit extracted from buyer i is at least:

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

By the second claim the profit of the dual it at least $\frac{c-1}{c-a_{\min}}$ times the cost of the primal, and thus, by weak duality theorem we conclude that the competitive ratio of the algorithm is $(1-R_{\max})\left(\frac{c-1}{c-a_{\min}}\right)$. □