Proceedings Article

Whole-page Optimization and Submodular Welfare Maximization with Online Bidders

NIKHIL R. DEVANUR, Microsoft Research, Redmond
ZHIYI HUANG, University of Pennsylvania
NITISH KORULA, Google Research, New York
VAHAB S. MIRROKNI, Google Research, New York
QIQI YAN, Google Research, Mountain View

In the context of online ad serving, display ads may appear on different types of web-*pages*, where each page includes *several* ad slots and therefore multiple ads can be shown on each page. The set of ads that can be assigned to ad slots of the same page needs to satisfy various pre-specified constraints including exclusion constraints, diversity constraints, and the like. Upon arrival of a user, the ad serving system needs to allocate a set of ads to the current web-page respecting these per-page allocation constraints. Previous slot-based settings ignore the important concept of a page, and may lead to highly suboptimal results in general. In this paper, motivated by these applications in display advertising and inspired by the submodular welfare maximization problem with online bidders, we study a general class of page-based ad allocation problems, present the first (tight) constant-factor approximation algorithms for these problems, and confirm the performance of our algorithms experimentally on real-world data sets.

A key technical ingredient of our results is a novel primal-dual analysis for handling free-disposal, which updates dual variables using a "level function" instead of a single level, and unifies with previous analyses of related problems. This new analysis method allows us to handle arbitrarily complicated allocation constraints for each page. Our main result is an algorithm that achieves a $1 - \frac{1}{e} - o(1)$ competitive ratio. Moreover, our experiments on real-world data sets show significant improvements of our page-based algorithms compared to the slot-based algorithms.

Finally, we observe that our problem is closely related to the submodular welfare maximization (SWM) problem. In particular, we introduce a variant of the SWM problem with online bidders, and show how to solve this problem using our algorithm for whole page optimization.

Categories and Subject Descriptors: F.1.2 [**Modes of Computation**]: Online Computation

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Display Ads, Primal Dual, Free Disposal

1. INTRODUCTION

With a multi-billion dollar market, display-related advertising – including banner ads, rich media, digital video and sponsorships – is a fast growing business that accounts for approximately 37% of Internet advertising [PwC and IAB 2011]. Unlike sponsored search advertising, display ads on the Internet are often sold in bundles of thousands or millions of impressions[1] over a particular time period. Advertisers pay the website publisher per impression and buy them ahead of time via contracts, often specifying a subset of pages

[1]The exposure of a user to a display ad on a web-page is called an "impression".

Part of this work is done while Z. Huang was an intern at Microsoft Research, Redmond. Z. Huang is supported by a Simons Graduate Fellowship for Theoretical Computer Science (No. 252128).

EC'13, June 16–20, 2013, Philadelphia, USA.

on which they would like their ads to appear, or a type of users they wish to target. The terms of these contracts may vary among advertisers and publishers but usually include a number of impressions to be assigned to a particular advertiser.

Display ad serving systems that assign ads to pages on behalf of web publishers must satisfy the contracts with advertisers, respecting targeting criteria and delivery goals. Modulo this, publishers try to allocate ads intelligently to maximize overall quality (measured, for example, by clicks). This has been modeled in the literature as an online allocation problem, where quality is represented by edge weights, and contracts are enforced by overall delivery constraints (e.g., [Feldman et al. 2009a; Mehta et al. 2007; Buchbinder et al. 2007]).

Display ads may appear on different types of *pages* (like sport, finance, or news sites) owned by a web publisher. In most cases, each page includes *several* ad slots and therefore *multiple ads can be shown on each page*. The set of ads that can be assigned to ad slots of the same page needs to satisfy various pre-specified constraints. One reason for this is that display ads are often used for *brand advertising*, in contrast to sponsored search ads, in which the goal is to get the user to take an immediate action. For example, when a user explicitly searches for "car rentals", both Hertz and Enterprise may wish for their ad to be shown (even and perhaps especially if their competitor's ad is shown, as they might otherwise lose a sale). On the other hand, when a user is viewing a sports website, Nike and Reebok might prefer that their ads not appear together. The set of constraints in display ads often includes (but not limited to):

— **Exclusion constraints**: Advertisers can have competitive relationships. One often needs to impose the constraint that if some slots are allocated to one advertiser, no slots are given to any of its competitors.
— **All-or-nothing constraints**: Some advertisers require that all or none of a set of related ads be shown on the same page. This is particularly common when ads reinforce each other.
— **Diversity constraints**: Publishers often want to diversify the ads shown to a user for each page. One way to do this is to form a hierarchical category of advertisers, and for each sub-category (possibly even containing a single advertiser) at each level, impose an upper bound on the number of impressions that can be allocated to advertisers within this sub-category.

As a result, the online optimization problem that the ad serving system must solve requires satisfying such complex page-level constraints. Previous research in online ad allocation and online matching ignores these important per-page constraints, and if applied directly to the page-based problem, may result in highly suboptimal outcomes. (It is easy to construct examples with either exclusion or all-or-nothing constraints with a competitive ratio that becomes worse linearly with the number of slots on a page.)

In this paper, we formally study page-based online ad allocation considering general allocation constraints with multiple ads per each page, and develop the first constant-factor competitive algorithms for these problems. In particular, assuming that the capacity of each ad is large, we develop a $1 - \frac{1}{e} - o(1)$-approximation for this problem. Furthermore, we show that our problems are closely related to the submodular welfare maximization (SWM) problem with online bidders, and our online algorithms also imply the same competitive ratio for the SWM problem with online bidders. Below, we first define these problems and summarize our results.

1.1. Problems and Results

In this paper, we define the *whole page optimization* problem (with free disposal). In this problem we have a finite set of advertisers A, and a finite set of online pages P, where each page consists of a (small) set I_p of impressions (or slots). For each page p, we have a family C_p of feasible alloctions, and for each feasible allocation $C \in C_p$ of page p, each

advertiser a may derive a value of $w_{p,C,a}$. An advertiser derives value from the top n_a best impressions she receives, where n_a is the number of impressions sold to her by contract. The only assumption about the C_p's is that we have access to a *demand oracle*: given a *cost* β_a for allocating an impression to each advertiser a, the demand oracle returns the configuration that maximizes the total value minus the total cost:

$$\arg\max_{C \in C_p} \left\{ \sum_a w_{p,C,a} - \beta_a n_{p,C,a} \right\},$$

where $n_{p,C,a}$ is the number of impressions allocated to a in configuration C. It is easy to construct polynomial-time demand oracles for most natural allocation constraints in this context like the exclusion, all-or-nothing, and diversity constraints described in the previous section. Note that we allow the value of an advertiser for an impression to depend on other ads shown in the page. Such a dependent-value model can model the fact that users' attention to a particular display ad on a page may depend on the whole set of ads on that page. Considerable research in advertising supports the idea that multiple ads in proximity affect how each ad is perceived; see, for instance, [Burke and Srull 1988; Mandese 1991; Keller 1991; Kent and Allen 1994] for such work in classical advertising, and [Athey and Ellison 2011; Aggarwal et al. 2008; Kempe and Mahdian 2008] for models for sponsored search ads.

In the online version of the problem, the pages arrive online one by one and a feasible allocation for a page must be chosen immediately upon its arrival. This choice cannot be changed later.

> As the main result in this paper, assuming that the capacities n_a are sufficiently large, we present a $1 - \frac{1}{e} - o(1)$-competitive algorithm for the whole page optimization problem.

This is also the optimal competitive ratio achievable. Without the assumption on large capacities n_a, the competitive ratio of our algorithm is 1/2. (See Section 3 for details). Further, our algorithms are eminently practical; we implemented and tested them on real data, and obtained improvements, over the algorithm of [Feldman et al. 2009a], (with different constraint levels) averaging 10 to 19%, and ranging up to 31 to 54%. See Section 5 for details.

Relationship to Online Submodular Welfare Maximization. Submodular Welfare Maximization is a well-studied problem in which a set V of items should be partitioned and allocated to a set A of bidders, each of whom has a submodular valuation function f_i; the goal is to maximize the total social welfare $\sum_{i \in A} f_i(V_i)$. The offline variant of this problem is well studied and it admits a $1 - \frac{1}{e}$-approximation algorithm [Vondrak 2008]. Commonly, the online version of the problem is concerned with the case where the items arrive online. In this paper, we propose a different version, where agents arrive online. In this online agent setting, given an offline set of items, bidders arrive online each with a monotone submodular (valuation) function over items. Upon arrival of each bidder, we assign an unconstrained subset of items to the bidder, allowing previously assigned items to be re-assigned to the current bidder. However, we may not assign or re-assign items to previous bidders. (This is in spirit similar to the literature on online allocations with buy-back [Feige et al. 2008; Constantin et al. 2009; Babaioff et al. 2009].) Our goal is to maximize welfare or total value of bidders at the end of the process. We show that the SWM problem with online bidders can be reduced to a special case of whole page optimization, and thus, we have the same competitive ratio for this problem. Moreover, if we have a *multiset* of items with many copies of each item, and no bidder wants more than a small number of copies of any item, the competitive ratio improves to $1 - \frac{1}{e} - o(1)$. One can also implement this algorithm in polynomial time given *demand oracle* access to the valuation functions. To the best of our

knowledge, this is the first competitive algorithm known for this natural online variant of the SWM problem.

1.2. Algorithm and Technique

The algorithm uses the primal-dual technique that has been used extensively for different generalizations of the online bipartite matching problem. The general format of such an algorithm is that it maintains a discount factor β_a for each advertiser a. For an allocation C, if advertiser a is assigned $n_{p,C,a}$ slots and receives total value $w_{p,C,a}$, we discount the value $w_{p,C,a}$ by an amount of $n_{p,C,a} \cdot \beta_a$. Formally, the format of the algorithm is as follows:

(1) Initially, $\beta_a = 0$ for each advertiser a.
(2) For every arriving page, do the following:
   (a) Choose feasible allocation C for the page maximizing the discounted value $\sum_a (w_{p,C,a} - n_{p,C,a} \cdot \beta_a)$
   (b) Allocate according to C.
   (c) Update β_a accordingly.

In order to define the final algorithm, it remains only to actually define the rule to update the discount factor β_a. The discount factors β_a's are interpreted as (a subset of) dual variables of a natural LP relaxation of the problem. The proof that such an algorithm is $1 - 1/e$ competitive requires two things, that the β_a's can be extended (by setting the remaining dual variables accordingly) to a feasible dual solution and that the total primal and dual objective values are within a factor of $1 - 1/e$.

Even though this technique has been used extensively, we offer new insights into the application of this technique, in particular for the class of *free disposal* problems, introduced by [Feldman et al. 2009a]. We first recount the state of the art in our understanding of this very important technique. Several variants of the online bipartite matching problem have had a $1 - 1/e$ competitive algorithm, albeit from seemingly different techniques. Some of the notable examples are as follows.

— The ranking algorithm of [Karp et al. 1990] for the online bipartite matching problem.
— A generalization of the ranking algorithm for the vertex weighted online bipartite matching problem, due to [Agarwal et al. 2011].
— The Adwords problem with small bids, due to [Mehta et al. 2007; Buchbinder et al. 2007], generalizing the online $b-$matching problem of [Kalyanasundaram and Pruhs 2000].
— The greedy algorithm for the Adwords problem with small bids and random arrival order, due to [Goel and Mehta 2008].

Recently [Devanur et al. 2013] gave a unification of all these results by showing how they all arise from essentially the same dual update function (which we call the exponential update function). They also showed how this update function and the competitive ratio of $1 - 1/e$ arise as the optimal solution to a particular differential equation. But the online matching with free disposal problem[2] of [Feldman et al. 2009a], which also had a $1 - 1/e$ competitive primal-dual algorithm, remained separate from this unification and seemingly used a different update rule.

> An important contribution of this paper is that we resolve this separation and show how the update function for [Feldman et al. 2009a] can be thought of as an extension of the exponential update function. Further this perspective allows us to naturally generalize this technique to the whole page optimization problem.

[2]In this problem, which is a precursor to the whole page optimization problem, *impressions* arrive online. Each impression i has a value w_{ia} for each advertiser a and advertiser a derives his value from the top n_a impressions assigned to him. Any extra impressions allocated to him are "disposed" off.

We now give a brief overview of how we achieve the above. We start with most basic problem, the online *fractional* bipartite matching problem, and the primal-dual analysis of the algorithm based on the exponential update function. The first step is a new primal-dual proof of the free disposal problem, extending the analysis of the fractional bipartite matching. The new idea needed for this as follows: the exponential update rule is based on the "level" of consumption for each advertiser. For the fractional matching problem, the level is the total fraction of edges matched to that vertex, for the Adwords problem, it is the fraction of the budget consumed by the advertiser. We extend the concept of level from being a real number to a real valued function from $\Re_+$ to itself. In other words, each real number $x \in \Re_+$ has its own level. This is because we also think of the *capacity* of an advertiser as a function from $\Re_+$ to itself. The capacity at x is the capacity to benefit from an impression of value x. Suppose that an advertiser has filled his capacity with impressions of value v. Then his capacity at x is filled for all $x \leq v$. However, he still has capacity to benefit from impressions of value $> x$. The level at x is simply the level to which the capacity at x is filled. The update rule is now the integral of the exponential update function of the level at x over the entire real line.

We next generalize the algorithm to the case where each impression could consume different amounts of the capacity of an advertiser. We show that instead of thinking of the level (and the capacity) as a function of the value, we should think of the level as a function of the *density*, which is the ratio of the value to the amount of capacity consumed by an impression. The algorithm and the analysis then extend naturally to this setting.

Finally we consider the whole page optimization problem. The main new difficulty here is that a particular allocation C that we have chosen for a page p could count towards one advertiser but not towards another. This issue does not arise in the earlier problems since each impression is allocated to only one advertiser. This change is captured in the LP relaxation by having one set of variables that capture the choice of C for each page and another set of variables that capture whether C is counted towards the capacity of each advertiser. Due to this, we have a new set of dual variables that need to be set and dual constraints that need to be satisfied. We show that these new dual variables have natural interpretations that allow us to extend the technique to this case. Another difference is that since an allocation now benefits multiple advertisers, we need to accumulate the "contributions" of all the advertisers to a given allocation in order to decide the best possible allocation. This is reflected in the way we choose the allocation, as the one maximizing the total discounted value among all feasible allocations.

1.3. Related Work

Our work is closely related to the previously studied online ad allocation problems, including the *Display Ads Allocation (DA)* problem [Feldman et al. 2009a; Feldman et al. 2010; Agrawal et al. 2009; Vee et al. 2010], and the *AdWords (AW)* problem [Mehta et al. 2007; Devanur and Hayes 2009]. In both of these problems, the publisher must assign online impressions to an inventory of ads, optimizing efficiency or revenue of the allocation while respecting pre-specified contracts. Both of these problems have been studied in the competitive adversarial model [Mehta et al. 2007; Feldman et al. 2009a; Buchbinder et al. 2007] and the stochastic random-arrival model [Devanur and Hayes 2009; Feldman et al. 2010; Agrawal et al. 2009; Vee et al. 2010].

The AW problem [Mehta et al. 2007; Buchbinder et al. 2007; Devanur and Hayes 2009] is related to our online allocation problem and the display ads allocation (DA) problem. In the AW problem, the publisher allocates impressions resulting from search queries. Advertiser j has a budget $B(j)$ on the total spend instead of a bound $N(j)$ on the number of impressions. Assigning impression i to advertiser j consumes $w(i, j)$ units of j's budget instead of 1 of the $N(j)$ slots, as in the DA problem. $1 - \frac{1}{e}$-approximation algorithms have been designed for this problem under the assumption of large budgets [Mehta et al. 2007; Buchbinder et al.

2007]. In the DA problem, given a set of m advertisers with a set S_j of eligible impressions and demand of at most $N(j)$ impressions, the publisher must allocate a set of n impressions that arrive online. Each impression i has value $w(i,j) \geq 0$ for advertiser j. The goal of the publisher is to assign each impression to one advertiser maximizing the value of all the assigned impressions. The adversarial online DA problem was considered in [Feldman et al. 2009a], which showed that the problem is inapproximable without exploiting *free disposal*; using this property (that advertisers are at worst indifferent to receiving more impressions than required by their contract), a simple greedy algorithm is $\frac{1}{2}$-competitive, which is optimal. When the demand of each advertiser is large, a $(1-\frac{1}{e})$-competitive algorithm exists [Feldman et al. 2009a], and this is tight. None of the previous work for the adversarial model consider the allocation of multiple ads per page, or general allocation constraints per page. Our primal-dual analysis is based on a new configuration linear program formulation as it needs to deal with an arbitrary family of allocation constraints per page, and therefore it is different from all the previous work.

Other than the adversarial model studied in this paper, online ad allocations have been studied extensively in various *stochastic models*. In particular, the problem has been studied in the *random order model*, where impressions arrive in a random order; and the *i.i.d.* model in which impressions arrive i.i.d. according to a known or an unknown distribution. There are two main category of algorithms used in such stochastic settings: *primal techniques* and *dual techniques*. The primal technique is based on solving an offline allocation problem on the instance that we expect to arrive according to the stochastic information, and then applying this offline solution online. This technique has been applied to the online stochastic matching problem [Karp et al. 1990] and in the i.i.d. model with known distributions [Feldman et al. 2009b; Menshadi et al. 2011; Haeupler et al. 2011], and resulted in improved competitive algorithms. The *dual technique* is based on computing an offline dual solution of an expected instance, and using this solution online [Devanur and Hayes 2009; Feldman et al. 2010; Agrawal et al. 2009; Vee et al. 2010]. Following the training-based dual algorithm of [Devanur and Hayes 2009], training-based $(1-\epsilon)$-competitive algorithms have been developed for the DA problem and its generalization to various packing linear programs [Feldman et al. 2010; Vee et al. 2010; Agrawal et al. 2009]. These papers develop a $(1-\epsilon)$-competitive algorithm for online stochastic packing problems in which $\frac{\text{OPT}}{w_{ij}} \geq O(\frac{m \log n}{\varepsilon^2})$ and the demand of each advertiser is large, in the random-order and the i.i.d model. It is not hard to generalize these techniques to capture the stochastic variant of the page-based ad allocation problem. Recently, improved approximation algorithms have been proposed for this problem [Karande et al. 2011; Mahdian and Yan 2011] in the random order model for unweighted graphs. Other than the above, online adaptive optimization techniques have been applied to online stochastic ad allocation [Tan and Srikant 2011; Devanur et al. 2011]. Such control-based adaptive algorithms achieve asymptotic optimality following an updating rule inspired by the primal-dual algorithms, but they do not achieve any bounded approximation factor for the adversarial model.

While these techniques provide improved approximation factors for stochastic models, they do not provide guaranteed approximations in the adversarial model. (However, this was achieved for the unweighted matching problem in [Mirrokni et al. 2011].) In reality, there are unexpected traffic spikes and dips and it is desirable to have an algorithm that can cope with such surprises. Our theoretical study of the whole page optimization problem in adversarial settings along with our experimental results for real-world data show that our algorithm satisfies these desirable properties.

2. ONLINE FRACTIONAL ASSIGNMENT

In this section, we describe $(1-1/e)$-competitive algorithms for the online weighted matching and online generalized assignment problems with free disposal. These results were previously

known from [Feldman et al. 2009a], but we analyze them here as a warm-up to the whole page optimization problem, and to demonstrate our unifying analysis. Key to our analysis is the Linear Program for weighted matching.

LP Formulation. Let x_{ia} be the indicator of allocating impression i to advertiser a. We will consider the following standard primal and dual linear programs of the online matching problem:

$$\begin{array}{lll} \text{Maximize } \sum_{i,a} w_{ia}x_{ia} & & \text{Minimize } \sum_a n_a\beta_a + \sum_i \alpha_i \\ \forall i: \quad \sum_a x_{ia} \le 1 & & \forall i,a: \quad \beta_a + \alpha_i \ge w_{ia} \\ \forall a: \quad \frac{1}{n_a}\sum_i x_{ia} \le 1 & & \forall i,a: \quad x_{ia}, \alpha_i, \beta_a \ge 0 \end{array} \tag{1}$$

2.1. Online bipartite fractional matching

A special case of the above problem is when the weights are either 0 or 1, and the capacity constraints are all 1. The instance can be thought of as a bipartite graph with advertisers on one side and impressions on the other, with an edge between them iff $w_{ia} = 1$. An allocation for such an instance is a matching in the bipartite graph. We actually consider fractional allocations here; a fractional allocation allows the allocation of an impression to an advertiser to be any real number in $[0, 1]$ with the sum of these "fractions" being no more than 1. The capacity constraints for the advertisers are that the sum of the fractions allocated to each advertiser is no more than his capacity. This is simply a solution to the LP relaxation (1). For this problem, one does not need free disposal since all the edge weights are 1. It is well known that there is a simple primal-dual algorithm for this problem with a $1 - 1/e$ competitive ratio. One of the goals of this paper is to point out how the algorithm and the analysis for the free disposal problem relates to that of the bipartite matching problem. We sketch a quick proof for the bipartite matching problem now. We use γ to denote the competitive ratio, which will be $1 - 1/e$. The algorithm builds primal and dual solutions so that

(1) The cost of the primal solution is at least γ times the cost of the dual solution.
(2) The dual constraint $\beta_a + \alpha_i \ge 1$ is feasible for all impressions seen so far.

It is easy to see that these two properties imply that the algorithm is γ-competitive.

We must now describe how to actually construct primal and dual solutions to satisfy these properties. In the beginning, all the primal and the dual variables are zero, hence primal and dual costs are both zero; thus the two properties are satisfied. The algorithm allocates impressions in a continuous process and we describe this process by specifying how primal and dual variables change as an infinitesimal quantity of an impression is allocated at any time. The dual variables, β_a's and α_i's are monotonically non-decreasing and are also changing continuously. When we allocate an infinitesimal quantity dx of an impression i to advertiser a, the primal cost increases by dx. The increase in the dual cost will then have to satisfy $d\beta_a + d\alpha_i \le dx/\gamma$; this maintains the invariant that the primal and dual costs are within a factor γ throughout, satisfying the first desired property.

Whenever there is an opportunity to allocate dx of impression i to an advertiser, there is up to dx/γ of the dual cost to go around. Different advertisers "offer" different ways to split this dual cost between the β_a's and α_i. The dx fraction of the impression is then allocated to the advertiser(s) who makes the highest offer for $d\alpha_i$. The offer made by an advertiser depends on the value of β_a he has already accumulated up to that point; this is because each advertiser tries to make sure that his own dual constraint is satisfied, that is $\beta_a + \alpha_i \ge 1$.

A lower β_a means that the advertiser needs to offer a higher amount. A natural choice (to help ensure dual feasibility, as we will show in Lemma 2.1) is to offer $d\alpha_i = (1 - \beta_a)dx$. To achieve $d\beta_a + d\alpha_i = dx/\gamma$, then, we must set $d\beta_a = (1/\gamma - 1 + \beta_a)dx$.

This differential equation in the dual variable β_a means that β_a will then be a function of the total fraction of impressions allocated to a, which is $y_a := \sum_i x_{ia}$. Let us denote the dependence of β_a on y_a as $\beta_a = G(y_a)$ for some monotonically non-decreasing function $G(\cdot)$. We denote the rate of change of β_a, $d\beta_a/dy_a$ by $g(y_a)$. That is, we can rewrite this equation as:

$$g(y) - G(y) = 1/\gamma - 1. \tag{2}$$

With this notation, at any point advertiser a offers to split dx/γ as $d\beta_a = g(y_a)dx$ and $d\alpha_i = (1-\beta_a)dx = (1-G(y_a))dx$. Equation (2) restricts our choice of $g(x)$ to be exponential; the particular functions we use are $g(x) = e^{x-1}/\gamma$ and $G(x) = \int_0^x g(y)dy = (e^{x-1} - e^{-1})/\gamma$. This gives $G(1) = 1$, which means that when $y_a = 1$, which is when the advertiser's capacity is exhausted, his offer is $1 - G(1) = 0$. This is by design, since once the advertiser's capacity is exhausted we don't want to allocate any more impressions to him. In fact, our choice of $G(\cdot)$ is the function that satisfies (2) for the largest possible constant γ, subject to the boundary constraint of $G(1) = 1$. With this background, we now define the algorithm:

ALGORITHM 1: Online bipartite fractional matching

1: Initialize all β_a's and α_i's to be zero.
2: **for** each impression i that arrives **do**
3: **while** $\sum_a x_{ia} < 1$ and $y_a < 1$ for some a s.t. $w_{ia} = 1$ **do**
4: Allocate a dx amount of i to each a in $\arg\max_{a:w_{ia}=1}\{1 - \beta_a\}$.
5: If dx of i is allocated to a, then increment β_a and α_i respectively by

$$d\beta_a = g(y_a)dx \quad \text{and} \quad d\alpha_i = (1 - \beta_a)dx \ .$$

6: **end while**
7: **end for**

The fact that this algorithm is γ-competitive follows from the two properties mentioned earlier, that the primal and the dual are within a factor of γ and that duals are feasible. The first follows immediately from (2) since whenever we allocate dx of i to a, the dual increase is $(g(y_a) + 1 - G(y_a))dx = dx/\gamma$. The proof that the second property also holds is as follows.

LEMMA 2.1. *For all a and i such that $w_{ia} = 1$, the dual variables β_a and α_i at the end of the algorithm are such that*

$$\beta_a + \alpha_i \geq 1.$$

PROOF. Consider the value of $y_a = \sum_i x_{ia}$ at the end of the algorithm. β_a at the end is equal to $G(y_a)$. If $y_a = 1$, then $\beta_a = G(1) = 1$ and $\alpha_i \geq 0$ and the lemma follows. Suppose that $y_a < 1$. Then the while loop for i must have ended with $\sum_a x_{ia} = 1$. Also, throughout the loop $d\alpha_i/dx$ must have been at least $1 - \beta_a$, since β_a is monotonically non-decreasing. Therefore $\alpha_i \geq 1 - \beta_a$. □

Since any feasible dual solution is an upper bound on the optimum offline solution, the competitive ratio follows.

THEOREM 2.2. *The algorithm is γ-competitive, with $\gamma = 1 - 1/e$.*

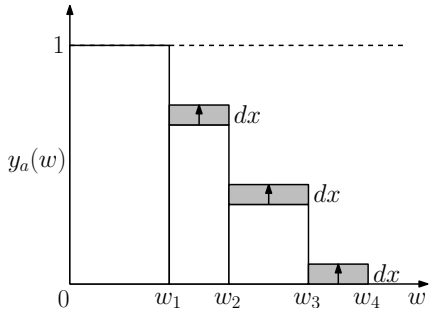

Fig. 1. Suppose advertiser a's capacity is fully occupied with minimal weight w_1. Further, some of the impressions have weight w_2 and some other have weight w_3. Then, allocating a dx amount of an impression with weight w_4 to a results in an increase in $y_a(w)$ by dx for $w_1 < w \leq w_4$.

2.2. Online Weighted Matching with Free Disposal

We now extend the algorithm and the analysis of the bipartite matching problem to the fractional version of the free disposal problem. One difference between the two problems is that in the free disposal problem, different impressions have different weights. Following the same framework as before, this means that the offer from advertiser a would depend on w_{ia}; since we have to ensure feasiblity of the dual constraint $\alpha_i + \beta_a \geq w_{ia}$, the natural choice is that advertiser a offers $d\alpha_i = (w_{ia} - \beta_a)dx$. Another difference is that the increase in the primal on allocating dx of impression i to a is not always $w_{ia}dx$, since a might have to discard some of its previously allocated impressions. If a discards dx of i' in order to accommodate dx of i, then the increase in the primal is $(w_{ia} - w_{i'a})dx$. The increment in β_a will then be $((w_{ia} - w_{i'a})/\gamma - w_{ia} + \beta_a)dx$. Now β_a is no more simply a function of $y_a = \frac{1}{n_a}\sum_i x_{ia}$. In fact, once $y_a = 1$, it stays there but β_a continues to increase as we allocate new impressions to a and discard older ones. β_a would be a function of the *history* of allocations to a, which would make the analysis rather complicated. We next present the key idea that makes sure that the analysis remains essentially the same as before, with a small extension.

Recall that earlier the offer of advertiser a was a function of y_a, which is the "level" to which his capacity was exhausted. The main idea, which allows this concept to be easily generalized to the free disposal problem (and beyond), is that instead of thinking of the level as a single number, we think of there being a level for every non-negative real number in $[0, \infty)$. Imagine that there is a capacity of 1 for each $w \in [0, \infty)$ and the level corresponding to w is the amount of capacity exhausted corresponding to w. (This is depicted in Figure 1.) To be precise, let

$$y_a(w) = \frac{1}{n_a} \sum_{i:w_{ia} \geq w} x_{ia}.$$

As we will show, we can define β_a to be a function of $y_a(\cdot)$; in fact it is the most natural generalization:

$$\beta_a = \int_0^\infty G(y_a(w))dw. \tag{3}$$

It is easy to see that in the unweighted case, this reduces to the earlier definition of $\beta_a = G(y_a)$. Suppose now that when an impression i arrives, a would have to discard i' in order to accept i. Here, i' is the impression with the smallest weight that is still allocated to a, i.e. $\arg\min_i\{w_{ia} : x_{ia} > 0\}$. If $\sum_i x_{ia} < 1$ then let i' be a dummy bidder with $w_{i'a} = 0$. If a dx amount of i is allocated to a, then the increase in the primal is $(w_{ia} - w_{i'a})dx$. One of the difficulties (*a priori*) with the free disposal problem is that the primal rate of increase

is $w_{ia} - w_{i'a}$ but the dual constraint is still $\beta_a + \alpha_i \geq w_{ia}$. With the definition of β_a as in (3) above, it is easy to see that this does not cause any problems. β_a has already accounted for the weight $w_{i'a}$ since $y_a(w) = 1$ for all $w \in [0, w_{i'a}]$ and

$$\int_0^{w_{i'a}} G(y_a(w))dw = \int_0^{w_{i'a}} 1dw = w_{i'a}.$$

With this observation, the offer of a to i, can be rewritten as

$$\frac{d\alpha_i}{dx} = w_{ia} - \beta_a = \int_{w_{i'a}}^{w_{ia}} (1 - G(y_a(w)))dw - \int_{w_{ia}}^{\infty} G(y_a(w))dw \leq \int_{w_{i'a}}^{w_{ia}} (1 - G(y_a(w)))dw.$$

The rate of increase in β_a w.r.t dx is

$$\frac{d\beta_a}{dx} = \int_{w_{i'a}}^{w_{ia}} g(y_a(w))dw$$

since $y_a(w)$ increases by dx precisely in the interval $[w_{i'a}, w_{ia}]$ and remains unchanged everywhere else. Therefore the total rate of increase in the dual cost is

$$\frac{d\alpha_i}{dx} + \frac{d\beta_a}{dx} \leq \int_{w_{i'a}}^{w_{ia}} \left(1 - G(y_a(w)) + g(y_a(w))\right) dw = \int_{w_{i'a}}^{w_{ia}} \frac{1}{\gamma} dw = \frac{w_{ia} - w_{i'a}}{\gamma}. \quad (4)$$

Therefore the primal and the dual rates of increase are within a factor of γ. With this, we now define the algorithm in Algorithm 2.

ALGORITHM 2: Online Weighted Matching with Free Disposal

1: Initialize all β_a's and α_i's to be zero.
2: For each a, create n_a dummy impressions with zero weight and allocate them completely to a.
3: **for** each impression i that arrives **do**
4: **while** $\sum_a x_{ia} < 1$ and $\beta_a < w_{ia}$ for some a **do**
5: Allocate a dx amount of i to each a in

$$\arg\max_{a: y_a(w_{ia}) < 1} \{w_{ia} - \beta_a\} \ .$$

6: If dx of i is allocated to a, then increment β_a and α_i respectively by

$$d\beta_a = \left(\int_{w_{i'a}}^{w_{ia}} g(y_a(w))dw \right) dx \quad \text{and} \quad d\alpha_i = (w_{ia} - \beta_a)\, dx \ ,$$

where for each a, the i' in the lower limit of the integral is in $\arg\min_i \{w_{ia} : x_{ia} > 0\}$.
7: Decrease $x_{i'a}$ by dx
8: **end while**
9: **end for**

As before we need to prove that primal and dual costs are within γ and dual feasibility. The proof goes along the same lines as before.

LEMMA 2.3. *The following are invariants throughout the algorithm.*

(1) *For all* a, $\frac{1}{n_a} \sum_i x_{ia} = 1$.
(2) *For all* a, *equation* (3) *holds.*
(3) *The primal and dual are within a factor of* γ.

PROOF. For (1), the statement is true initially due to the allocation of the dummy impressions. Subsequently whenever we allocate a dx amount of an impression i to a, we

discard an equal amount of another impression i'. Therefore the invariant is maintained throughout.

For (2), suppose Eq. (3) holds before the arrival of impression i. Consider a step in the algorithm where dx of i is allocated to a for some i, and an equal amount of i' is discarded. w_{ia} is strictly greater than $w_{i'a}$, since otherwise $\beta_a \geq w_{ia}$ by Eq. (3). Note that for all $w \in [0, w_{i'a}]$, $y(w) = 1$ by the definition of i' and $y(w)$ does not change due to the step. Also for all $w \in (w_{ia}, \infty]$, $y(w)$ does not change. Finally, for all $w \in (w_{i'a}, w_{ia}]$, $y_a(w)$ increases by dx.

Recall that in this step β_a is incremented by $d\beta_a = \left(\int_{w_{i'a}}^{w_{ia}} g(y_a(w))dw\right) dx$. By the above argument, this increment can be written as

$$d\beta_a = \int_0^\infty \Big(g(y_a(w))dy_a(w)\Big)dw = \int_0^\infty dG(y_a(w))dw \ .$$

Therefore Eq. (3) continues to hold.

Proof of (3): We already argued this and proved it in Eq. (4). □

LEMMA 2.4. *For all a and i, the dual variables β_a and α_i at the end of the algorithm are such that*

$$\beta_a + \alpha_i \geq w_{ia}.$$

PROOF. Consider the value of $y_a(\cdot)$ at the end of the algorithm. If $\beta_a \geq w_{ia}$, then by $\alpha_i \geq 0$ the lemma follows. If $\beta_a < w_{ia}$, then the while loop for i must have ended with $\sum_a x_{ia} = 1$. Throughout the loop $d\alpha_i/dx$ must have been at least $w_{ia} - \beta_a$, since β_a is monotonically non-decreasing. Therefore $\alpha_i \geq w_{ia} - \beta_a$. □

THEOREM 2.5. *Algorithm 2 is γ-competitive, with $\gamma = 1 - 1/e$.*

3. CONFIGURATIONS

We now consider a generalization of the problem where multiple advertisements can be shown on a single page. It is the pages which arrive online, instead of impressions as before. A page has multiple distinct slots in which ads can be placed and a configuration of ads for a page specifies which ad is shown in each slot. The same ad (or ads from the same advertiser) may be shown on multiple slots on the same page. There may be rules about which configurations are allowed; for each page p we denote the set of feasible configurations for that page by $\mathcal{C}_p$. The value derived by an advertiser may depend not only on where his own ads are shown, but also on which ads are shown in the other slots. In other words it depends on the entire configuration of ads. For a configuration $C \in \mathcal{C}_p$ the value derived by advertiser a is denoted by $w_{p,C,a}$.

For an advertiser a, the number of different slots his ad is shown in configuration C on page p is denoted by $n_{p,C,a}$. We also refer to this as the number of impressions allocated. Advertiser a has a bound n_a on the total number of impressions that can be allocated to him. The free disposal version of this is that he can be allocated more impressions, but we only count the top n_a impressions towards the objective function. A given configuration may be counted towards one advertiser and be not counted towards another. The configurations are picked online and cannot be changed later, but the accounting of which impressions to count towards an advertiser may be changed. In particular, the algorithm adds new impressions to this pool of top n_a impressions and drops some of the ones picked earlier. Once an impression is dropped it is never considered again.

Let $c_{p,C,a} := n_{p,C,a}/n_a$. The following is an LP relaxation for the above problem, and its dual. The variable $z_{p,C}$ indicates whether configuration C is chosen for page p. The variable $x_{p,C,a}$ indicates whether the impressions in configuration C on page p are counted towards

the top n_a impressions for advertiser a.

$$\text{Maximize} \sum_{p,C,a} w_{p,C,a} \cdot x_{p,C,a} \qquad \text{Minimize} \sum_a \beta_a + \sum_p \alpha_p$$

$$\forall a: \quad \sum_{p,C} c_{p,C,a} \cdot x_{p,C,a} \leq 1 \qquad \forall p, C, a: \quad \delta_{p,C,a} + c_{p,C,a} \cdot \beta_a \geq w_{p,C,a}$$

$$\forall p, C, a: \quad x_{p,C,a} \leq z_{p,C} \qquad \forall p, C: \quad \alpha_p \geq \sum_a \delta_{p,C,a}$$

$$\forall p: \quad \sum_{C \in \mathcal{C}_p} z_{p,C} \leq 1 \qquad \forall p, C, a: \quad x_{p,C,a}, z_{p,C}, \alpha_p, \beta_a, \delta_{p,C,a} \geq 0$$

As before we will consider the fractional version of the problem, which is just a solution to the LP above. This can be easily extended to the integral version when the n_a's are all large.

The main new component introduced by this problem is that multiple advertisers can benefit from one configuration. The concept of offers used earlier extends naturally here: each advertiser makes an offer for each of the configurations, based on his value for the configuration and his level function. The configuration chosen is simply the one for which the sum of the offers of all the advertisers is the highest. Another new aspect in this problem is that there are new dual variables, namely $\delta_{p,C,a}$. These simply capture the offer made by each advertiser for each configuration and don't appear in the objective function. Except for these small modifications the algorithm and the analysis is almost identical to the previous case with densities.

When a page p arrives, suppose we were to allocate dx of the page to the configuration $C \in \mathcal{C}_p$. Then each advertiser potentially gets an additional value of $w_{p,C,a}dx$, if this configuration was better for him than the ones he already has. For a given advertiser a, suppose he would have to discard dx' of a previously allocated configuration C' on page p' in order to accept C. Let $\rho_{p,C,a} := w_{p,C,a}/c_{p,C,a}$; then $(p', C') = \arg\min_{p,C}\{\rho_{p,C,a} : x_{p,C,a} > 0\}$. The actual increase in the primal objective value corresponding to a due to this allocation is then

$$w_{p,C,a}dx - w_{p',C',a}dx' = (\rho_{p,C,a} - \rho_{p',C',a}) \cdot c_{p,C,a}dx. \tag{5}$$

This increase in the primal is split between the dual variables based on the function $y_a(\cdot)$ which as before is defined as follows:[3]

$$y_a(\rho) = \sum_{p,C:\rho_{p,C,a} \geq \rho} c_{p,C,a} \cdot x_{p,C,a}.$$

β_a is defined in terms of y_a as before:

$$\beta_a = \int_0^\infty G(y_a(\rho))\, d\rho. \tag{6}$$

Now a offers an amount of $\delta_{p,C,a} = w_{p,C,a} - c_{p,C,a} \cdot \beta_a$ to each configuration C from which he can benefit, i.e., each C such that $y_a(\rho_{p,C,a}) < 1$. Otherwise, she offers $\delta_{p,C,a} = 0$. We allocate dx amount of a given page p to the configuration C that receives the highest total offer. The dual variable α_p is then incremented by $\sum_a \delta_{p,C,a}dx$. We can bound $\delta_{p,C,a}$ as

$$\delta_{p,C,a} = w_{p,C,a} - c_{p,C,a} \cdot \beta_a = c_{p,C,a}(\rho_{p,C,a} - \beta_a) \leq c_{p,C,a} \int_{\rho_{p',C',a}}^{\rho_{p,C,a}} (1 - G(y_a(\rho)))\, d\rho.$$

[3]Note that configuration C is beneficial to advertiser a iff $y_a(\rho_{p,C,a}) < 1$. We will use this notation to filter only those advertisers for whom a configuration is actually beneficial.

since $y_a(\rho_{p',C',a}) = 1$ and $G(y_a(\rho)) = 1$ for $\rho \in [0, \rho_{p',C',a}]$.

The increase in β_a is

$$\frac{d\beta_a}{dx} = c_{p,C,a} \int_{\rho_{p',C',a}}^{\rho_{p,C,a}} g(y_a(\rho))d\rho$$

since $y_a(\cdot)$ remains unchanged everywhere except in $(\rho_{p',C',a}, \rho_{p,C,a}]$, and for all ρ in that interval $y_a(\rho)$ increases by $c_{p,C,a}dx$. The total rate of increase in dual is

$$\frac{d\alpha_p}{dx} + \sum_a \frac{d\beta_a}{dx} \leq \sum_a c_{p,C,a} \int_{\rho_{p',C',a}}^{\rho_{p,C,a}} \left(1 - G(y_a(\rho)) + g(y_a(\rho))\right) d\rho = \sum_a \frac{c_{p,C,a}(\rho_{p,C,a} - \rho_{p',C',a})}{\gamma},$$

which is $1/\gamma$ times the primal rate of increase, from (5).

The entire algorithm is summarized as Algorithm 3.

ALGORITHM 3: Free disposal with configurations

Initialize all primal and dual variables to be zero.
Create a dummy page p with a single configuration C such that for all $a, w_{p,C,a} = 0$ and $c_{p,C,a} = 1$. Set $x_{p,C,a} = z_{p,C} = 1$.
for each page p that arrives **do**
  while $\sum_C z_{p,C} < 1$ and $c_{p,C,a} \cdot \beta_a < w_{p,C,a}$ for some a and C **do**
    Allocate a dx amount of p to each C (that is, increase $z_{p,C}$ by dx) in

$$\arg\max_C \left\{ \sum_a \max \left\{ 0, w_{p,C,a} - c_{p,C,a} \cdot \beta_a \right\} \right\} .$$

    if dx of p is allocated to C **then**
      Increment α_p by

$$d\alpha_p = \textstyle\sum_{a: c_{p,C,a} \cdot \beta_a < w_{p,C,a}} \left(w_{p,C,a} - c_{p,C,a} \cdot \beta_a \right) dx.$$

      Increase $x_{p,C,a}$ by dx.
      for each a such that $c_{p,C,a} \cdot \beta_a < w_{p,C,a}$ **do**
        Increment β_a by

$$d\beta_a = c_{p,C,a} \left(\textstyle\int_{\rho_{p',C',a}}^{\rho_{p,C,a}} g(y_a(\rho))d\rho \right) dx ,$$

        where the p', C' is in $\arg\min_{p,C}\{\rho_{p,C,a} : x_{p,C,a} > 0\}$.
        Decrease $\rho_{p',C',a}$ by $c_{p,C,a}dx/c_{p',C',a}$.
      end for
    end if
  end while
  Set $\delta_{p,C,a} = \max\{0, w_{p,C,a} - c_{p,C,a} \cdot \beta_a\}$.
end for

We need to prove that primal and dual costs are within γ and dual feasibility.

LEMMA 3.1. *The following are invariants throughout the algorithm:*

(1) for all a, $\sum_{p,C} c_{p,C,a} \cdot x_{p,C,a} = 1$.
(2) for all a, equation (6) holds.
(3) The primal and dual are within a factor of γ.

The proof is similar to that of Lemma 2.3 and hence omitted.

LEMMA 3.2. *For all a, p and C, the dual variables β_a, α_p and $\delta_{p,C,a}$ at the end of the algorithm are such that*

(1) $c_{p,C,a}\beta_a + \delta_{p,C,a} \geq w_{p,C,a}$ *and*
(2) $\alpha_p \geq \sum_a \delta_{p,C,a}$.

PROOF. Consider the first constraint. If for some p, C, a, it so happens that $c_{p,C,a} \cdot \beta_a < w_{p,C,a}$, then it is satisfied by the definition of $\delta_{p,C,a}$ and the fact that β_a is monotonically non-decreasing. Otherwise, $\delta_{p,C,a} = 0$ but $c_{p,C,a}\beta_a \geq w_{p,C,a}$ so the first constraint is still satisfied.

Consider the second constraint for a given page p. Suppose that the while loop for that page ends in $c_{p,C,a} \cdot \beta_a \geq w_{p,C,a}$ for all a and C. Then $\delta_{p,C,a} = 0$ for all a, C and hence the second constraint is trivially satisfied.

Now suppose that the while loop for p ended with $\sum_a z_{p,C} = 1$. Throughout the loop $d\alpha_a/dx$ only decreases since β_a's are all monotonically non-decreasing. Further, $\max_C \sum_a \delta_{p,C,a}$ is exactly the value of $d\alpha_a/dx$ at the end of the while loop. Therefore the second constraint is satisfied at the end of the while loop. These dual variables are not changed after that, so it continues to hold. □

THEOREM 3.3. *Algorithm 3 is γ-competitve, with $\gamma = 1 - 1/e$.*

4. SUBMODULAR WELFARE MAXIMIZATION WITH ONLINE BIDDERS

In this section, we consider a variant of the Submodular Welfare Maximization (SWM) problem. Here, items are known offline, and bidders arrive online; this is contrast to the more well-studied online variant where bidders are known offline and items arrive online. In our problem, at every time step, a bidder arrives with a monotone submodular function over items. We then assign an unconstrained subset of items to the bidder, allowing previously assigned items to be assigned again. However if an item was assigned to a previous bidder, but is now assigned to a new bidder, the old bidder is no longer assigned the item. Our goal is to maximize welfare or total value of bidders at the end of the process.

Note that for this online SWM to make sense, we need to allow one-way reassignment of items since otherwise, no reasonable competitive ratio can be achieved for this problem. Also it is worth noting that such a reassignment is in spirit similar to the literature on buy-back [Feige et al. 2008; Constantin et al. 2009; Babaioff et al. 2009], except that we can buy back for free.

In the following, we show that SWM with online bidders can be reduced to the whole page optimization setting. In making the connection, the intended meaning of bidders and items in the context of whole page optimization will be reversed. In particular, items now correspond to offline advertisers, and bidders now correspond to online pages.

LEMMA 4.1. *Given a ρ-competitve algorithm for whole page optimization for arbitrary $n'_a s$, there is a ρ-competitive algorithm for SWM with online bidders.*

PROOF. Given an instance of SWM with online bidders, we construct a corresponding whole page optimization setting as follows. Let there be m items numbered $1, \ldots, m$. For each item j, there is a corresponding advertiser j with capacity one. For each bidder with a monotone submodular function $f(\cdot)$ over the item set, we construct a page p with m slots in the following way. For each subset of items $S \subseteq \{1, \ldots, m\}$, include a feasible allocation configuration where for all $j \in S$ slot j is assigned to advertiser j, and all slots outside S are not assigned. Furthermore, the value of advertiser j in this configuration is defined as $f(\{1, \ldots, j\} \cap S) - f(\{1, \ldots, j-1\} \cap S)$. Note that the values are defined in a way such that the total value of allocated advertisers in S is equal to $f(S)$, and it follows that the offline versions of both the SWM problem and the whole page optimization problem have identical solutions and optimal values.

Now given a ρ-approximation algorithm for whole page optimization, we can simulate it on the above whole page optimization instance using a demand oracle. If an allocation is chosen, which specifies the set of advertisers that get assigned, then for each such advertiser,

say j, in the online SWM problem we assign the corresponding item j to the current bidder either if (1) item j wasn't assigned before, or if (2) the value by doing this is higher than the value v of item j for the bidder that it was assigned to previously. In the latter case, let b be the bidder that was assigned the item j, in whole page optimization, we lose a value of v in accounting for advertiser j, while in the online SWM problem, by submodularity, bidder b loses a value of at most v. It follows that at the end of process, the algorithm for online SWM achieves total objective value that is at least as large as the algorithm for whole page optimization. Since both problem settings share the same optimal value, our lemma follows. □

Our algorithm for whole page optimization can give a $\frac{1}{2}$-approximation even when capacities of advertisers are small, by setting $d\alpha_i = d\beta_a = w_{ia}dx$. It follows that we have a $\frac{1}{2}$-approximation algorithm for SWM with online bidders. Furthermore, under the following assumption, whole page optimization gives a $1 - \frac{1}{e} - o(1)$-approximation for this problem: Consider a more general setting where the item set is a multi-set, and submodularity is defined w.r.t. multi-sets. At every step, the arriving bidder reports a monotone submodular valuation function defined on the items. For this setting, we can apply our result for whole page optimization to get $(1 - \frac{1}{e} - o(1))$-approximation assuming that the minimum multiplicity of an item tends to infinity.

Furthermore, our whole page optimization algorithm can be implemented in polynomial time given demand oracle access. (Details in full version.)

5. EMPIRICAL EVALUATION

An important motivation behind the whole page optimization problem is the display ad allocation with whole-page-based constraints. Besides being theoretically optimal, a key feature of our algorithm is its simplicity and ease of implementation, allowing easy empirical evaluation. In this section, we present experimental results, comparing a whole-page allocation algorithm to the slot-based equivalent.

Experimental Details. Our data sets consist of impressions for 5 (anonymous) publishers from 2 days in January 2012. The number of daily impressions per publisher varies from roughly 150,000 to 1,300,000, and the number of advertisers *per publisher* is up to several hundred. Advertisers specify complex targeting criteria to define the set of eligible impressions (giving the bipartite graph between impressions and advertisers), and the *edge weights* capture the "targeting quality" (in these experiments, click probability) of an advertiser for an impression. The specification of all per-page constraints for each advertiser is non-trivial and hard to describe succinctly; in fact, many internet advertising services specify constraints differently. We do not describe all the nuances of the ad-serving system's constraints, as our goal here is to demonstrate that significant improvements are possible by considering configurations for the entire page at once. Therefore, we present results here for the case of only exclusion constraints (where advertiser a can specify that their ad is not to be shown along with the ad of competitor b); further, to aid reproducibility of these experiments, we consider *randomly generated* pairwise exclusions. From the point of view of the online algorithm, the manner in which exclusions are generated is irrelevant; the algorithm simply works with the graph specifying which pairs of ads cannot be shown together. That is, we work with "real" weighted bipartite graphs between impressions and advertisers (as in previous work [Feldman et al. 2010]), but use randomly generated per-page constraints. This allows us to (a) demonstrate that the significant improvements obtained are not due to specific constraints of the advertisers for these publishers, and (b) investigate how the performance of the algorithm changes with an increase in the number of constraints.

In every other respect, the experimental setup is as close to a real system as possible; impressions are considered by the algorithms in the order of the page-views of the cor-

responding users to the publishers' websites. We work with the real capacities / budget constraints of advertisers, etc.

A separate issue is that in real systems, the algorithms used can be stochastic, or based on historical data. One could repeat our experiments with a single-slot-at-a-time stochastic algorithm vs. a page-configuration-based stochastic algorithm. We do not report on such experiments here, as this paper is focused on worst-case algorithms. A further advantage of worst-case algorithms is that they can cope with new advertisers, changing capacities, etc.; for this reason, algorithms used in practice are typically a hybrid of worst-case and stochastic algorithms; for more details, see [Feldman et al. 2010].

Algorithms. The algorithms we used are essentially similar to those of this paper and the slot-based algorithm of [Feldman et al. 2009a], with a few minor differences: Our theoretical results assumed that an impression could be infinitesimally split among multiple advertisers; instead, we discretize the algorithm by assigning an impression i to a single advertiser a in $\arg\max\{w_{ia} - \beta_a\}$, breaking ties arbitrarily. Further, the ideal discretization might have a each one of a million impressions contributing in a slightly different way to β_a, all of which must be updated after each allocation; we bucket these, but this does not significantly affect algorithm performance; this also helps deal with floating-point issues. For the page-based algorithm, recall that we need demand oracle access to find the optimal configuration for a page; in practice, we explicitly solve a (small) integer program to find the optimal configuration that satisfies the exclusion constraints. Though such an integer program could, in general, require time exponential in the number of page slots, the number of slots (and advertisers eligible for each slot) is typically quite small. Generally accepted latency to serve an ad request is on the order of 50-100 ms; the authors have in fact used a (specialized) integer program solver to find the optimal configuration for 'real' instances with even more complex constraints in far less time than this.

Results. For each publisher, we inserted random exclusion constraints between advertisers with varying probabilities. Since these were the only page-level constraints considered, at a constraint probability of 0, the two algorithms (page- and slot-based) are identical. Table 5 shows the performance of the algorithms on each publisher with constraint probabilities ranging from 0.1 to 0.3. As one might expect, the performance of both algorithms decreased (monotonically) with an increase in the constraint probabilities. Note, though, that the decrease as a function of constraint probability is *much* more significant for the slot-based algorithm than the page-based one, an average of 16% vs. 4.6%. (Figure 2 illustrates this for 1 publisher). In fact, for 3 out of the 5 publishers, the page-based allocation performance decays so slowly that the score of the page-based algorithm with constraint probability 0.3 is *higher* than the slot-based algorithm with probability 0.1.

Overall, we note a significant gain from using page-based allocation, going from an average of 3.9% with constraint probability 0.1 to an average of 18.6% with constraint probability 0.3. There is, of course, considerable variation among publishers; at a constraint probability of 0.2, the gain from using page-based allocation ranges from 3.88% to 31.08%, and at a constraint probability of 0.3, the gain ranges from 9.32% to 53.93%.

Further Discussion. We note that page-based allocation is of even more importance when the publisher's inventory of impressions is almost fully sold to advertisers. If there is a surplus of users (many more than required by the contracts sold in advance), the deficiencies of a slot-based algorithm are less significant; even if it makes sub-optimal decisions, leaving several slots empty to satisfy page-level constraints, it can "make up the difference" with the surplus users. Those ads under-assigned to the first users can be shown to those arriving later; the surplus of users ensures that there are enough high-quality impressions for each advertiser. On the other hand, if there are few users, it is critically important that early opportunities not be wasted, and page-based algorithms have an even clearer advantage.

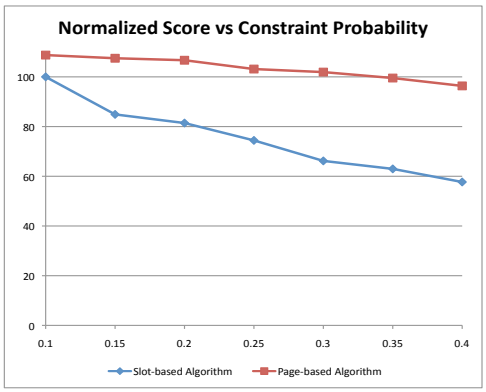

Fig. 2. Performance vs constraint probability, Publisher B

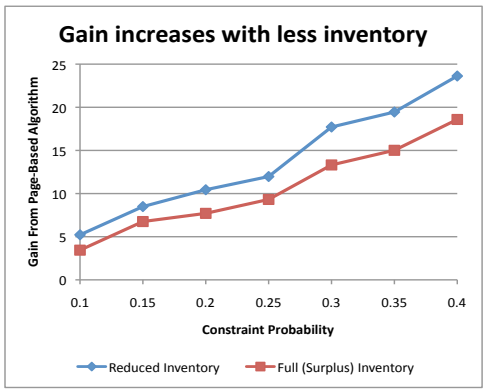

Fig. 3. Increased gain with reduced inventory, Publisher D

	Pub A		Pub B		Pub C		Pub D		Pub E		Avg		Gain
Prob.	Slot	Page	Slot	Page	Slot	Page	Slot	Page	Slot	Page	Slot	Page	
0.1	100	102.7	100	108.8	100	100.8	100	103.4	100	103.9	100	103.9	3.9%
0.15	98.7	102.1	84.9	107.5	97.0	99.3	94.4	100.8	98.4	103.5	94.7	102.6	8.3%
0.2	96.9	101.7	81.4	106.7	94.1	97.9	93.3	100.5	96.5	103.2	92.4	102.0	10.4%
0.25	94.9	101.2	74.4	103.1	89.3	96.0	91.4	99.9	95.5	103.0	89.1	100.6	12.9%
0.3	92.3	100.9	66.2	101.9	82.6	94.5	86.6	98.2	93.1	102.5	84.0	99.6	18.6%

Normalized scores comparing the slot-based and page-based algorithms for each publisher, and averaged over all publishers. Scores are normalized for each publisher such that the slot-based algorithm with constraint probability 0.1 has a score of 100. The average column is a simple average, not weighted by the number of impressions per publisher.

We demonstrate this by repeating the experiments for the 5 publishers above, randomly sampling half the users. As one can see from Figure 3 for Publisher D, the benefit of page-based allocation is larger for these reduced-inventory instances than in the original instances. Even the publisher with least gain (Publisher C) sees its gain go from 3.88% to 5.36% at the constraint probability of 0.2. In general, using our algorithm for whole page optimization produces high single-digit to double-digit percentage gains compared to the slot-based algorithms, and for supply-constrained publishers, we see gains of another 3-5%.

The experiments above only considered exclusion constraints; these play a particularly significant role in small or niche websites, where many of the advertisers may compete with each other to target a particular community of users. For many publishers, *all-or-nothing* (sometimes referred to as road-blocking) constraints are also important. It is clear that page-based allocation plays an important role here as well; if a slot-based algorithm picks an ad with a 5-or-nothing constraint for one slot, it is compelled to pick the ad 4 more times on the page, regardless of how low a "targeting quality" or weight the ad may have for those 4 slots. Other kinds of constraints are also used in practice, but these vary from one publisher to another, and it is harder to compare these scientifically and publish results of reproducible experiments.

REFERENCES

Agarwal, G., Goel, G., Karande, C., and Mehta, A. 2011. Online vertex-weighted bipartite matching and single-bid budgeted allocation. In *SODA*. SIAM.

Aggarwal, G., Feldman, J., Muthukrishnan, S., and Pál, M. 2008. Sponsored search auctions with markovian users. *WINE*, 621–628.

Agrawal, S., Wang, Z., and Ye, Y. 2009. A dynamic near-optimal algorithm for online linear programming. *Computing Research Repository*.

Athey, S. and Ellison, G. 2011. Position auctions with consumer search. *The Quarterly Journal of Economics 126,* 3, 1213–1270.

Babaioff, M., Hartline, J., and Kleinberg, R. 2009. Selling ad campaigns: Online algorithms with cancellations. In *EC*. 61–70.

Buchbinder, N., Jain, K., and Naor, J. S. 2007. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA*. Springer, 253–264.

Burke, R. and Srull, T. 1988. Competitive interference and consumer memory for advertising. *Journal of Consumer Research*, 55–68.

Constantin, F., Feldman, J., Muthukrishnan, S., and Pal, M. 2009. Online ad slotting with cancellations. In *SODA*. 1265–1274.

Devanur, N. and Hayes, T. 2009. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *EC*. 71–78.

Devanur, N., Jain, K., and Kleinberg, R. 2013. Randomized primal-dual analysis of ranking for online bipartite matching. In *SODA*.

Devanur, N. R., Jain, K., Sivan, B., and Wilkens, C. A. 2011. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *EC*. ACM, 29–38.

Feige, U., Immorlica, N., Mirrokni, V., and Nazerzadeh, H. 2008. A combinatorial allocation mechanism for banner advertisement with penalties. In *WWW*. 169–178.

Feldman, J., Henzinger, M., Korula, N., Mirrokni, V. S., and Stein, C. 2010. Online stochastic ad allocation: Efficiency and fairness. Tech. rep.

Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., and Pal, M. 2009a. Online ad assignment with free disposal. In *WINE*.

Feldman, J., Mehta, A., Mirrokni, V., and Muthukrishnan, S. 2009b. Online stochastic matching: Beating 1 - 1/e. In *FOCS*. 117–126.

Goel, G. and Mehta, A. 2008. Online budgeted matching in random input models with applications to adwords. In *SODA*. 982–991.

Haeupler, B., Mirrokni, V., and ZadiMoghaddam, M. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*. 170–181.

Kalyanasundaram, B. and Pruhs, K. R. 2000. An optimal deterministic algorithm for online b -matching. *Theoretical Computer Science 233,* 1–2, 319–325.

Karande, C., Mehta, A., and Tripathi, P. 2011. Online bipartite matching with unknown distributions. In *STOC*. 587–596.

Karp, R., Vazirani, U., and Vazirani, V. 1990. An optimal algorithm for online bipartite matching. In *STOC*. 352–358.

Keller, K. 1991. Memory and evaluation effects in competitive advertising environments. *Journal of Consumer Research*, 463–476.

Kempe, D. and Mahdian, M. 2008. A cascade model for externalities in sponsored search. In *WINE*. Springer, 585–596.

Kent, R. and Allen, C. 1994. Competitive interference effects in consumer memory for advertising: The role of brand familiarity. *The Journal of Marketing*, 97–105.

Mahdian, M. and Yan, Q. 2011. Online bipartite matching with random arrivals: A strongly factor revealing lp approach. In *STOC*. 597–606.

Mandese, J. 1991. Rival spots cluttering tv. *Advertising Age 18*.

Mehta, A., Saberi, A., Vazirani, U., and Vazirani, V. 2007. Adwords and generalized online matching. *J. ACM 54,* 5, 22.

Menshadi, V. H., Oveis Gharan, S., and Saberi, A. 2011. Online stochastic matching: Online actions based on offline statistics. In *SODA*. 1285–1294.

Mirrokni, V. S., Oveis Gharan, S., and ZadiMoghaddam, M. 2011. Simultaneous approximations of stochastic and adversarial budgeted allocation problems. In *SODA*. 1690–1701.

PwC and IAB 2011. IAB Internet advertising revenue report, 2011. PricewaterhouseCoopers and the Interactive Advertising Bureau. http://www.iab.net/media/file/IAB-HY-2011-Report-Final.pdf.

Tan, B. and Srikant, R. 2011. Online advertisement, optimization and stochastic networks. In *CDC-ECC*. IEEE, 4504–4509.

Vee, E., Vassilvitskii, S., and Shanmugasundaram, J. 2010. Optimal online assignment with forecasts. In *EC*. 109–118.

Vondrak, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*. 67–74.