Page-based Ad Allocation and Submodular Welfare Maximization with Online Bidders

Nitish Korula, Google Research, New York. Email: nitish@google.com
Vahab Mirrokni, Google Research, New York. Email: mirrokni@google.com
Qiqi Yan, Stanford University. Email: contact@qiqiyan.com

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 certain 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.

In particular, we study page-based online ad allocation in two variants of *independent-value setting under matroid constraints* and a *dependent-value setting with arbitrary constraints*. These settings allow us to model complicated allocation constraints for each page, as well as how an advertiser's value is affected by the presence of other ads. For both settings, we study a simple algorithm that optimizes for each page with the well-matched advertisers suitably discounted, and our main result is that this algorithm achieves $1-\frac{1}{e}-o(1)$ 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 both variants of our problem are 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 the general dependent value setting. This reduction is done by employing a cross-monotonic value sharing scheme for submodular functions.

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 on which they would like their ad to appear, or a type of user 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

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

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 to be satisfied by display ads often includes (but is not limited to):

- **Exclusion constraints**: Ads from competing companies should not be displayed on the same page.
- **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**: There may be an upper bound on the number of ads on a single page that can be shown from one advertiser.

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 equal to the 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 the number of ads per page is a constant and the capacity of each ad is large, we develop a $1 - \frac{1}{e} - o(1)$-approximation for this problem in the presence of an downward-closed[2] family of allocation constraints per page. Furthermore, we show that our problems are closely related to the submodular welfare maximization (SWM) problem with online items or 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 two variants of the page-based online ad allocation problem: an *independent-value variant with matroid constraints*, and a *dependent-value model with arbitrary constraints*. The first model is a special case of the second model, and it enables an easier analysis of our algorithm, and serves as a warm-up to the more general and abstract dependent-value model in Section 4. In both models, 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). We consider general allocation constraints of multiple ads per page, and develop the first constant-factor competitive algorithms for these problems.

In Section 3, we study an illustrative special case of our main problem, called the *independent-value variant with matroid constraint*. (We abbreviate this as PA-Indep, for Page Allocation with Independent Values.) In this case, the family of feasible subsets of slots on a page to be assigned to each advertiser forms a matroid. For example, in the presence of a k-uniform matroid constraint, at most k ads of each advertiser can be shown on a page. (The matroid for each advertiser can be different.) Moreover, each

[2] A family F of subsets is downward closed if for any feasible set S in F, all subsets of S are also in F.

advertiser a has a value and weight $w_{a,i,p}$ for each impression i on a page p, and she derives value from the top n_a best impressions she receives, where n_a is the number of impressions sold to her by contract. For this problem, we present an algorithm that achieves a $1 - \frac{1}{e} - o(1)$-approximation if the capacities $n(a)$ are sufficiently large. In this setting, using the matroid intersection algorithm as a subroutine, our algorithm can be executed efficiently in polynomial time in the number of impressions per page.

Next, as our main contribution, we study PA-Dep, a general *dependent-value model with general constraints* with value-sharing. We allow arbitrary constraints over feasible sets of ads for each page, imposing only the requirement that feasible allocations are downward closed w.r.t. advertisers. Such families of feasible allocations include the most natural allocation constraints in this context like the exclusion, all-or-nothing, and diversity constraints described in the previous section. We also 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 2012; Aggarwal et al. 2008; Kempe and Mahdian 2008] for models for sponsored search ads. Thus, we consider the most general setting in which advertisers share the value of a page, and the value each advertiser gets is affected by what other advertisers are allocated to this page. More formally, 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-share of $v_p(C, a)$, where v_p is a value-sharing function. 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 general PA-Dep problem for any family C_p of downward-closed allocations, and any *cross-monotone*[3] value sharing functions v_p. Without the assumption on large capacities n_a, the competitive ratio of our algorithm is $1/2$. (See Section 4 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 with a submodular valuation function f_i, and 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]. Both the PA-Indep and PA-Dep problems are closely related to different variants of the online submodular welfare maximization (SWM) problem. The well-known SWM problem with online items is a generalization of the easier PA-indep problem. This implies a simple $\frac{1}{2}$-competitive algorithm for the PA-Indep problem (see Section 3). Here, we study the *SWM problem with online bidders*: 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 later. 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 PA-Dep, and thus, we have the same competitive ratio for this problem. In particular, 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 $1 - \frac{1}{e} - o(1)$. To prove that an online competitive algorithm for the PA-Dep problem gives an online competitive algorithm for SWM with online bidders with the same guarantee, we use the fact that submodular functions ad-

[3] In fact, we need a weaker notion of cross-monotonicity for this. See Section 5 for more details.

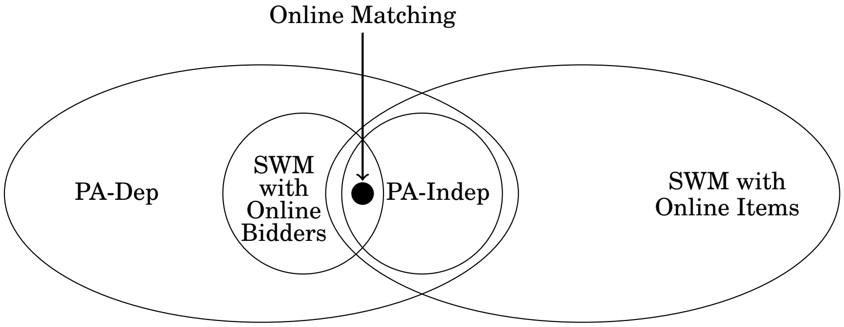

Fig. 1. Problems studied in this paper, and their relations.

mit a cross-monotonic value sharing method [Moulin and Shenker 2001]. 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

In allocation problems where items arriving online must be allocated to a set of agents known in advance, a central issue is that we do not want to have an agent receive too many items early on, as in the future, there might be many items that are good exclusively for this agent. As is common in algorithms for such problems, we handle this issue by suitably *discounting* agents who have received many items. In particular, our algorithm maintains a discount factor β_a for each agent/advertiser a; we describe the process for computing β_a later in this section. More precisely, for an allocation C, if advertiser a is assigned t_a slots and receives total value $v_a(C)$, we discount the value $v_a(C)$ by an amount of $t_a \cdot \beta_a$, where β_a is a suitably defined (exponentially-weighted) average of all the weights of slots assigned to a so far. This is a generalization of [Feldman et al. 2009a], where at most one slot is assigned at every time step. If the value for each advertiser is determined solely by the slots where her ad is shown, it is not entirely surprising that the approach of [Feldman et al. 2009a] can be generalized, though there are some subtle technical details to be considered. What is more interesting is that this approach can also be made to work even when the values for advertisers depend on which other ads are shown on the page, and in which slots.[4] This separability of discount factors is surprising, given that the total value for an advertiser is not simply a function of the slots she receives; we discuss this further in Section 4.

Formally, our algorithm PD-Exp is defined as follows:

(1) Initially, $\beta_a = 0$ for each advertiser a.
(2) For every arriving page, do the following:
    (a) Choose a feasible allocation C to maximize the discounted value $\sum_a (v_a(C) - t_a \cdot \beta_a)$
    (b) Allocate according to C.
    (c) Recalculate β_a as defined below.

In order to define the final algorithm, it remains only to actually define the rule to compute the discount factor β_a. β_a is computed as an exponentially-weighted average of all the weights of slots assigned to a so far. The analysis of this algorithm for the general PA-Dep problem is based on a primal-dual analysis of a new configuration LP

[4] This requires mild assumptions, detailed in Section 4

formulation for this problem. See Sections 2 and 4 and for the details of the algorithm and the analysis.

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, 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 AdWords (AW) problem [Mehta et al. 2007; Buchbinder et al. 2007; Devanur and Hayes 2009] is related to our online allocation problem and the display ad allocation(DA) problem. In the AdWords (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 has 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 it is tight. None of the previous work for the adversarial model consider the allocation of multiple ads per page, and general allocation constraints per page. Our primal-dual analysis is based on a new configuration linear programming formulation as they need to deal with an arbitrary family of allocation constraints per each page, and therefore it is different than 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 *iid* model in which impressions arrive iid according to a known or an unknown distribution. There are two main main category of algorithms used in such stochastic settings: *primal techniques* and *dual techniques*. The primal technique is based on solving offline allocation problem on an 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] 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 algorithm. The *dual technique* is based on computing an offline dual solution of an expected instance, and use this solution online [Devanur and Hayes 2009; Feldman et al. 2010; Agrawal et al. 2009; Vee et al. 2010]. Following a training-based dual algorithm by Devanur and Hayes [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 2010; 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, [Mirrokni et al. 2012] achieves this for the unweighted matching problem.) Desirable algorithms should be able to cope with unexpected traffic spikes and dips seen in reality. Our theoretical study of the page-based online allocation problem problem in adversarial settings along with our experimental results for real-world data show that our algorithms satisfy these desirable properties for the more general page-based allocation problem.

2. DEFINING THE EXPONENTIALLY-WEIGHTED AVERAGE

In order to complete the definition of the PD-Exp algorithm described in Section 1.2, we need to define the exact exponentially-weighted average function used to update the discount factors. The following notation will be useful in defining the exponentially-weighted average discount factor β_a: Given an advertiser a, let a have capacity n_a and let d_a be the maximum number of slots that a can get in a page. We let $r_a = n_a/d_a$ denote the capacity ratio of advertiser a; this is the minimum number of pages that can be used to satisfy the contract of a.

We let multiplier $\alpha_a = (1+\frac{1}{r_a})^{\frac{1}{d_a}}$, and let $e_x = (1+\frac{1}{x})^x$. Finally, we choose $\hat{n}_a$ to be the minimum of $\frac{\left(e_{r_a}\cdot(1+\frac{1}{r_a})^{-1}\cdot\alpha_a\right)-1}{(e_{r_a}-1)} \cdot \frac{1}{\alpha_a-1}$ and $\frac{e_{r_a}-\alpha_a^{d_a-1}}{e_{r_a}-1} \cdot n_a$. Finally let $\rho_a = \frac{\hat{n}_a(e_{r_a}-1)}{n_a \cdot e_{r_a}}$. We will be omitting subscripts a when they are clear from context.

Fix an advertiser a. Note that fixing d, we have $\alpha = 1+o(1)$, $e_r = (1-o(1))e \rightarrow 2.718\ldots$, $\hat{n} = (1-o(1))n$, and $\rho = (1-o(1))\cdot(1-\frac{1}{e})$, where the $o(1)$ terms go to 0 as the capacity ratio r goes to infinity. When $d = n = 1$, we have $\alpha = 2$, $r = 1$, $e_r = 2$, $\hat{n} = 1$, and $\rho = \frac{1}{2}$.

Definition 2.1 (*Exponentially Weighted Average Scoring*). Fix an advertiser a with capacity $n = n_a$. Let $w_1 \geq w_2 \geq \ldots \geq w_n$ be the top n weights assigned to a (pad with zero weights if fewer than n weights are assigned to a), and let $d \in \{1,\ldots,n\}$, then the exponentially weighted average score (subsequently abbreviated exp-avg) of a is defined as $\beta_a = \frac{1}{\hat{n}\cdot(e_r-1)} \cdot \sum_{i=1}^{n} \alpha^{i-1} \cdot w_i$.

The normalizing coefficient in front is chosen such that if all the top n weights are equal to w_n, then the exp-avg score will be $(1+o(1))\cdot w_n$. (Note if we replaced $\hat{n}$ in the denominator with n, we would get exactly w_n, giving a true weighted average.) This slight deviation from the semantics of "average" turns out to be crucial for technical reasons. The main property of exp-avg that will be useful for us is the following lemma, which is a nontrivial generalization of a lemma in [Feldman et al. 2009a]. It allows the update of multiple weights at a time, which imposes a number of technical challenges.

LEMMA 2.2. *Fix an advertiser a with capacity $n = n_a$. Let β_{old} be the exp-avg score of a. Suppose $t \in \{1, \ldots, d\}$. Let $v_1 \geq \ldots \geq v_t$ be the $(n-t+1)$-th to n-th highest weights already assigned to a, and we assign t new weights $u_1 \geq \ldots \geq u_t$ to a where $u_i \geq \beta_{old}$ for all i. Let β_{new} be the recalculated exp-avg score thereafter. Then the following hold:*

(1) *The new weights $u_1, \ldots, u_t$ replace $v_1, \ldots, v_t$ as part of the top n weights.*
(2) $\sum_{i=1}^{t}(u_i - v_i) \geq \rho \cdot n \cdot (\beta_{new} - \beta_{old}) + \rho \cdot \sum_{i=1}^{t}(u_i - \beta_{old})$.

PROOF. To prove the first claim, it suffices to show that $\beta_{old} \geq v_1$. To see this, all the top $n-t+1$ weights, and in particular all the top $n-d+1$ weights are at least v_1, and (recalling that $r = n/d$) it follows that:

$$\beta_{old} \geq \frac{1}{\hat{n}(e_r-1)} \cdot \sum_{i=1}^{n-d+1} \alpha^{i-1} \cdot v_1 \;=\; \frac{1}{\hat{n}(e_r-1)} \cdot \frac{\alpha^{n-d+1}-1}{\alpha-1} \cdot v_1$$
$$= \frac{(1+\frac{1}{r})^{\frac{n}{d}-1}\alpha - 1}{e_r - 1} \cdot \frac{1}{\hat{n}(\alpha-1)} \cdot v_1 \;=\; \frac{e^r \cdot (1+\frac{1}{r})^{-1} \cdot \alpha - 1}{e_r - 1} \cdot \frac{1}{\hat{n}(\alpha-1)} \cdot v_1 \geq v_1,$$

where the last inequality is by our choice of $\hat{n}$.
To prove the second claim, we first consider the extreme case that $u_1, \ldots, u_t$ become the very top t weights. When this is the case, observe that:

$$\beta_{new} \;=\; \beta_{old} \cdot \alpha^t + \frac{1}{\hat{n} \cdot (e_r-1)} \cdot \sum_{i=1}^{t} u_i \alpha^{i-1} - \frac{1}{\hat{n} \cdot (e_r-1)} \cdot \sum_{i=1}^{t} v_i \cdot \alpha^{n+i-1}$$
$$\leq \; \beta_{old} \cdot \alpha^t + \frac{\alpha^{d-1}}{\hat{n} \cdot (e_r-1)} \cdot \sum_{i=1}^{t} u_i - \frac{\alpha^n}{\hat{n} \cdot (e_r-1)} \cdot \sum_{i=1}^{t} v_i.$$

Note that $\alpha^t = (1+\frac{d}{n})^{\frac{t}{d}} \leq 1 + \frac{t}{n}$ for $t \leq d$. It follows that:

$$n(\beta_{new} - \beta_{old}) + \sum_{i=1}^{t}(u_i - \beta_{old})$$
$$\leq ((\alpha^t - 1) \cdot n - t)\beta_{old} + \left(\frac{n\alpha^{d-1}}{\hat{n}(e_r-1)} + 1\right) \cdot \sum_{i=1}^{t} u_i - \left(\frac{n \cdot e_r}{\hat{n}(e_r-1)}\right) \cdot \sum_{i=1}^{t} v_i \;\leq\; \frac{1}{\rho} \cdot (\sum_{i=1}^{t} u_i - \sum_{i=1}^{t} v_i).$$

Here the last line is because we chose $\hat{n}$ such that the coefficient of $\sum_{i=1}^{t} u_i$ is no bigger than the coefficient of $\sum_{i=1}^{t} v_i$, and that we defined $\frac{1}{\rho}$ to be the coefficient of $\sum_{i=1}^{t} v_i$.
Next we consider the case that $u_1, \ldots, u_t$ do not all make it to the top t weights. (but they will still be among the top n weights by the first claim) Therefore there exists some $i = 1, \ldots, t$ and some k such that u_i is the k-th largest weight, while the $(k-1)$-th largest with value z is not one of $u_1, \ldots, u_t$.

Fixing the values of u_i's and the values of all weights previously associated with advertiser a except z. Consider lowering the value of z to be equal to u_i, and let u_i win the tie-breaking and become the $(k-1)$-th largest. The left hand side of the target inequality is unchanged, and we argue that the right hand side can only increase.

Note that the u_i terms are again unaffected. The term of $n\beta_{new}$ decreases by an amount of $\frac{n}{\hat{n}(e_r-1)}$ times $\alpha^{k-1} \cdot (z - u_i)$, while the term of $-(n+t) \cdot \beta_{old}$ increase by an amount of $\frac{n+t}{\hat{n}(e_r-1)}$ times $\alpha^j(z - u_i)$, where j is the rank of weight z before $u_1, \ldots, u_t$ were added. Note that $j \geq k - 1 - t$. Therefore $\alpha^{k-1} \leq \alpha^t \alpha^j = \frac{n+t}{n} . \alpha^j$, and hence the total increase is as much as the total decrease. It follows that the right hand side can only increase.

Now we repeatedly lower old weights this way until all u_i's are among the top t weights. This results in the hardest case of the problem, as the target inequality has the same left hand side and a larger right hand side. As we already took care of this case, our proof is complete. $\square$

3. THE INDEPENDENT VALUE MODEL WITH MATROID CONSTRAINTS

In this section, we describe the independent value model with matroid constraints, which we call PA-Indep. This setting enables an easier analysis of PD-Exp, and serves as a warm-up to the more general and abstract dependent-value model in Section 4. Moreover, for problems in this setting, PD-Exp can be executed very efficiently. In particular, the running time of PD-Exp for each page is polynomial in both the number of advertisers and the number of impressions in this page.

3.1. Model

In the formal model, we have a finite set of advertisers A, and a finite set of pages P, where each page $p \in P$ consists of a finite number of impressions I_p (also called slots). For simplicity, we assume that each advertiser has a single type of ad to show.

Allocation For each page p, an assignment specifies for each slot which advertiser's ad to show, and the feasibility of an assignment is specified by one matroid set system for each advertiser (see [Oxley 1992] for an introduction to matroids). In particular, each advertiser a is associated with a matroid set system $\mathcal{M}_{p,a}$, and we say that an assignment is feasible if and only if for each advertiser a, the set of slots assigned to a is an independent set in $\mathcal{M}_{p,a}$. We use $\mathcal{C}_p$ to denote the set of all feasible assignments for page p.

Matroids The use of matroid set systems allow us to model various types of allocation constraints. For example, if every advertiser is associated with a 1-uniform matroid[5], then feasible assignments essentially correspond to matchings, imposing the constraint that no ad is shown more than once in a page. More generally, a web page can have different types of ad slots, including slots that are in a right column, or slots that are in between text. Partition matroids[6] can allow us to specify for each advertiser and each type of slots, a limit on how many ads can be shown for the type.

Value We assume that each advertiser a has a value or weight $w_{p,a,i}$ for each impression i in page p. Furthermore, each advertiser a only derives value from the n_a best impressions she receives, where n_a is called the capacity of advertiser a.

Online Process The online allocation process is the following. At the beginning, the set of advertisers is revealed, along with their capacities. At every time step, a page of slots arrives, along with all the incident weights (i.e., the weights of every advertiser for every slot in this page) and the feasibility constraint for the page. Our algorithm then must immediately assign slots in this page to advertisers (possibly leaving some unassigned), subject to the given feasibility constraint.

3.2. Algorithm and Primal-Dual Analysis

In the context of PA-Indep, the PD-Exp algorithm chooses a feasible allocation C to maximize $\sum_{(a,i)\in C}(w_{p,a,i} - \beta_a)$. Discount scores β_a are then reevaluated to be the ex-

[5] In a k-uniform matroid over n elements, a subset is feasible if and only if its cardinality is at most k.

[6] In a partition matroid over a ground set X, a partitioning $X_1, \ldots, X_t$ of X is given along with numbers l_i for $1 \leq i \leq t$. A subset of X is feasible if and only if its intersection with X_i has size at most l_i for all $1 \leq i \leq t$.

ponentially weighted average after the assignments. Our main theorem of this section is that PD-Exp gives a $(1 - \frac{1}{e} - o(1))$ -approximation for the PA-Indep-Matroid setting.

THEOREM 3.1. *For the online page-based ads allocation problem with independent values and matroid constraints, PD-Exp gives a $(1 - \frac{1}{e} - o(1))$-approximation to the offline optimal value, as the minimum capacity ratio of an advertiser goes to infinity.*

We remark that PD-Exp can be executed efficiently for our setting. For each page, the optimization problem can be cast as maximizing total weight subject to intersection of two matroid constraints, which can be done in polynomial time (see [Schrijver 2003]). To see this, consider a ground set over all (advertiser, slot) pairs. The matroid constraints for different advertisers are over disjoint parts of the ground set, and their union is again a matroid constraint. On the other hand, the constraint that each slot can go to at most one advertiser can be captured by a partition matroid.

To prove Theorem 3.1, we use a primal-dual LP analysis.

Let $r_{p,a}(S)$ denote the rank function of the matroid $\mathcal{M}_{p,a}$ associated with page p and advertiser a. Let $x_{p,a,i}$ indicate whether advertiser a derives value from the i-th impression of page p. Consider the following linear program: (square brackets enclose the dual variables)

$$\begin{array}{rcl} \text{maximize} & \sum_{a,p,i} w_{p,a,i} \cdot x_{p,a,i} & \textbf{(Primal)} \\ \forall p,i: & \sum_a x_{p,a,i} \le 1 & [z_{p,i}] \\ \forall p,a,S: & \sum_{i \in S} x_{p,a,i} \le r_{p,a}(S) & [\gamma_{p,a,S}] \\ \forall a: & \sum_{p,i} x_{p,a,i} \le n_a & [\beta_a] \\ \forall p,a,i: & x_{p,a,i} \ge 0 & \end{array}$$

Here the first constraint encodes that at most one advertiser can derive value from an impression of a page. The second constraint encodes that the set of impressions in a page that an advertiser derives value from actually satisfies the matroid constraint. The third constraint encodes that each advertiser can derive value from at most n_a impressions.

This linear program serves as a linear relaxation to the offline problem. Therefore, its value gives an upper-bound on the optimal offline objective value.

The corresponding dual linear program is also useful to us:

$$\begin{array}{rcl} \text{minimize} & \sum_{p,i} z_{p,i} + \sum_{p,a,S} r_{p,a}(S) \cdot \gamma_{p,a,S} + \sum_a n_a \cdot \beta_a & \textbf{(Dual)} \\ \forall p,a,i: & z_{p,i} + \sum_{S: i \in S} \gamma_{p,a,S} + \beta_a \ge w_{p,a,i} & [x_{p,a,i}] \\ \forall p,a,i,S: & z_{p,i}, \gamma_{p,a,S}, \beta_a \ge 0 & \end{array}$$

In the following we show how to derive feasible primal and dual solutions from the execution of PD-Exp, so that the value of the primal solution equals the value of the algorithm, and that the value of the primal solution is at least $1 - \frac{1}{e} - o(1)$ fraction of the value of the dual solution. Our theorem follows as the value of the dual solution upper-bounds the optimal primal value by weak LP duality, which then upper-bounds the value of the optimal solution.

Initially we set all primal and dual variables to zero. Consider the running of PD-Exp for page p. The algorithm finds the allocation C that maximizes $\sum_{(a,i) \in C} (w_{p,a,i} - \beta_a)$ subject to the matroid constraints and the constraint that each slot goes to at most one advertiser, where β_a is the exp-avg score of advertiser a right before the arrival of page p. This maximization problem for page p can be captured by the following "local" linear program:

$$\begin{array}{rcl}
\text{maximize} & \sum_{a,i}(w_{p,a,i} - \beta_a) \cdot \hat{x}_{a,i} & \text{(Primal)} \\
\forall i: & \sum_a \hat{x}_{a,i} \le 1 & [\hat{z}_i] \\
\forall a, S: & \sum_{i \in S} \hat{x}_{a,i} \le r_{p,a}(S) & [\hat{\gamma}_{a,S}] \\
\forall a, i: & \hat{x}_{a,i} \ge 0 &
\end{array}$$

In particular, as the underlying constraint is given by the intersection of two matroids, the discounted value of allocation C exactly equals to the value of this local LP.

The corresponding local dual LP is the following:

$$\begin{array}{rcl}
\text{minimize} & \sum_i \hat{z}_i + \sum_{a,S} r_{p,a}(S) \cdot \hat{\gamma}_{a,S} & \text{(Dual)} \\
\forall a, i: & z_i + \sum_{S: i \in S} \hat{\gamma}_{a,S} \ge w_{p,a,i} - \beta_a & [\hat{x}_{a,i}] \\
\forall a, i, S: & \hat{z}_i, \hat{\gamma}_{a,S} \ge 0 &
\end{array}$$

Given optimal solutions from these local LPs, we update primal and dual variables for the original "global" LPs as follows:

- For all a, i, S, set $x_{p,a,i} = \hat{x}_{a,i}$, $z_{p,i} = \hat{z}_i$, and $\gamma_{p,a,S} = \hat{\gamma}_{a,S}$.
- For each advertiser a,
  - If the number of nonzero entries of $x_{p,a,i}$ exceeds the capacity n_a, pick the non-zero $x_{p,a,i}$ variables with the lowest $w_{p,a,i}$ coefficients, and set their values to 0.
  - Re-evaluate β_a to be the exponential weighted average score of advertiser a. Let $\beta_a^{old}, \beta_a^{new}$ be the scores before and after re-evaluating, respectively.

To see that both primal and dual variables are feasible, note that whenever the variables are set for a page, feasibility constraints are satisfied. Later on, β_a variables can only increase, and previous $x_{p,a,i}$ variables can only decrease, neither of which can affect feasibility.

Next we relate the increase in primal objective (*primal gain*) to the increase in dual objective (*dual gain*).

For every page p, the dual gain is $\sum_i z_{p,i} + \sum_{a,S} r_{p,a}(S) \cdot \gamma_{p,a,S}$ plus the total increase in β_a variables. Here the first term equals to $\sum_i \hat{z}_i + \sum_{a,S} r_{p,a}(S) \cdot \hat{\gamma}_{a,S}$, which then equals to $\sum_{a,i}(w_{p,a,i} - \beta_a) \cdot \hat{x}_{a,i}$ by strong LP duality. Therefore the dual gain from each advertiser a with capacity n_a is $w_{p,a,i} - \beta_a^{old}$ for each slot i assigned to a, plus the increase in β_a which equals $n_a(\beta_a^{new} - \beta_a^{old})$. For the primal objective, let $u_1, \ldots, u_t$ for $t \le d$ be the weights of impressions assigned to advertiser a. They are all as high as β_a^{old}. By Lemma 2.2, $u_1, \ldots, u_t$ become part of the top n_a weights for a. Let $v_1, \ldots, v_t$ be the replaced weights. So the primal gain is $\sum_{i=1}^t u_i - \sum_{i=1}^t v_i$. Again by Lemma 2.2, $\sum_{i=1}^t (u_i - v_i) \ge \rho_a \cdot n_a(\beta_a^{new} - \beta_a^{old}) + \rho_a \cdot \sum_{i=1}^t (u_i - \beta_a^{old})$. It follows that dual gain is at least ρ_a fraction of primal gain for every page.

Finally, summing over all advertisers and all pages, we have that the final dual objective is at least $\min_a \rho_a$ fraction of the final primal objective, completing the proof.

Remark 3.2. One can also define the PA-Indep-DC setting, where the feasibility constraint for each advertiser is given by a downward-closed set system instead of a matroid. One can prove essentially the same guarantee, but using the different (more general) analysis in Section 4.

3.3. Connection to Submodular Welfare Maximization with Online Items

Problems in the PA-Indep setting can be reduced to the standard online submodular welfare maximization problem. In the standard version of online submodular welfare maximization problem, we have offline bidders as well as online items that arrive sequentially one by one. Upon the arrival of an item, we assign it to one of the bidders immediately and irrevocably. Each bidder's valuation for a set of items is monotone and submodular, and our goal is to maximize welfare, or the total values of bidders. In the context of PA-Indep, bidders correspond to advertisers, and items correspond to slots.

PA-Indep is connected to SWM with online bidders in the following way. See Appendix A for the proof and some further discussion on the connection.

LEMMA 3.3. *Given a ρ-approximation algorithm for SWM with online bidders, there is a ρ-approximate algorithm for PA-Indep-Matroid.*

It is known that a simple greedy algorithm [Lehmann et al. 2001; Nemhauser and Wolsey 1981] is a $\frac{1}{2}$-approximation algorithm for online SWM. It follows that there is a $\frac{1}{2}$-approximation algorithm for PA-Indep-Matroid. It is open whether a $(1-\frac{1}{e})$-approximation algorithm exists for online SWM, the existence of which could improve Theorem 3.1.

4. A DEPENDENT VALUE MODEL BASED ON VALUE-SHARING

In this section, we study PA-Dep, a general dependent value model based on value-sharing. Such a model allows the value of an advertiser for an impression to depend on other ads shown on the page, which can model, for example, the fact that different ads compete for the user's attention. The main result of this section is that PD-Exp also has competitive ratio $1-\frac{1}{e}-o(1)$ for this general setting, albeit via a different primal-dual LP analysis.

4.1. Model

In the formal model, we again have a finite set of advertisers A, and a finite set of online pages P, where each page consists of a (small) set of impressions I_p (or slots).

Allocation For each page p, an allocation specifies for each impression i in page p, which advertiser it is assigned to, or if it is not assigned at all. In contrast with the previous section, we allow very general constraints. For each page p, the set of feasible allocations is given by a non-empty set $\mathcal{C}_p$ that is downward-closed w.r.t. advertisers, in the sense that if an allocation is feasible, if we restrict this allocation to a subset of advertisers, the resulting allocation is also feasible.

Value-Sharing We think of advertisers as "sharing" the page, and the value each advertiser gets is affected by what other advertisers are allocated in this page. Formally, for each feasible allocation $C \in \mathcal{C}_p$ of page p, each advertiser a can derive a value-share of $v_p(C,a)$. We let $v_p(C,a)$ be zero if a is not assigned in C. We make the following cross-monotonicity assumption on value-sharing, which says that for an allocation C, if we remove one advertiser from the allocation, the remaining advertisers as a whole are better off. Formally, a value-sharing function $v_p(C,a)$ is cross-monotonic, if for all C, a, we have $\sum_{a' \neq a} v_p((C \backslash a), a') \geq \sum_{a' \neq a} v_p(C, a')$, where $C \backslash a$ denote the allocation obtained from C by removing assignments for advertiser a. Note that this cross-monotonicity assumption is weaker than the standard cross-monotonicity condition in the cost-sharing literature [Moulin and Shenker 2001].

Final Value If an advertiser a gains a value-share of v from an allocation of a page where she is assigned n slots, we think of a as being assigned n slots each having weight of $\frac{v}{n}$. Each advertiser a can only derive value in the best possible way from at most n_a slots. If the advertiser receives more than n_a slots in total, the excess slots of minimum value do not count towards her total.

Online Process At the beginning, the advertisers and their capacities are revealed to us. At each time step, a page p arrives with the set of slots I_p, the feasible allocations $\mathcal{C}_p$, and the value-sharing function v_p. We then choose a feasible allocation from C_p immediately and irrevocably.

The PA-Dep setting is very general in two main aspects. First, the only requirement we impose is that the set of feasible allocations be downward closed w.r.t. advertisers; this gives us great flexibility in modeling various real life constraints. In particular, all of the constraints described in the introduction can be captured:

— **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:** Advertisers can require that all or none of a specified set of related ads appear on a page; this is often used for a set of ads which 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 at each level, impose an upper-bound on the number of impressions that can be allocated to advertisers within this sub-category.

Second, the cross-monotonicity assumption is a weak condition satisfied in natural scenarios. This condition is trivially satisfied in the PA-Indep setting of Section 3. Here for an allocation C, the value share $v_p(C, a)$ of an advertiser a is the total weight of slots assigned to a in C; clearly the removal of an advertiser does not affect the value of other advertisers in the allocation. This also shows that PA-Indep is a special case of the PA-Dep setting.

It is challenging to precisely model how advertisers derive value from a shared page. Several attempts have been made to capture such effects (in particular, see [Kempe and Mahdian 2008; Aggarwal et al. 2008; Athey and Ellison 2012] for such modeling for sponsored search ads). In this paper, we abstract out these issues to focus on the online allocation problem, by assuming that the value-sharing function $v_p(\cdot)$ is given to us; our algorithm works with any such model.

4.2. PD-Exp and Main Theorem

For an allocation C, let C_a denote the set of slots assigned to advertiser a. In the context of PA-Dep, the PD-Exp algorithm for each page chooses an allocation C to maximize $\sum_{a \in C} (v_p(C, a) - |C_a| \cdot \beta_a)$, and allocates accordingly. Our main result of this section is that PD-Exp gives a $(1 - \frac{1}{e} - o(1))$ -approximation for the PA-Dep setting.

THEOREM 4.1. *For the online page-based ads allocation problem with cross-monotonic value-sharing, PD-Exp gives a* $(1 - \frac{1}{e} - o(1))$*-approximation to offline optimal, as the minimum capacity ratio of an advertiser goes to infinity. Further, when every advertiser has a capacity and capacity ratio of 1, the approximation ratio is* $\frac{1}{2}$.

In general, PD-Exp may not run in time polynomial in the number of slots. However, in practice, the number of slots in a page is usually a small constant on the order of 3-10. Therefore, we expect PD-Exp to have reasonable running time in practice.

4.3. Primal Dual Analysis

We prove Theorem 4.1 using a primal dual analysis different from that of Theorem 3.1.

Let $x_{p,C,a} \in \{0, 1\}$ denote whether advertiser a derives value from allocation $C \in \mathcal{C}_p$ on page p. The primal linear program is the following:

$$\text{maximize } \sum_{p, C \in \mathcal{C}_p, a} v_p(C, a) \cdot x_{p,C,a} \quad \text{(Primal)}$$

$$\begin{array}{rcl} \forall p, a: & \sum_{C \in \mathcal{C}_p} x_{p,C,a} \leq 1 & [z_{p,a}] \\ \forall a: & \sum_{p, C \in \mathcal{C}_p} |C_a| \cdot x_{p,C,a} \leq n_a & [\beta_a] \\ \forall p, C \in \mathcal{C}_p, a: & x_{p,C,a} \geq 0 & \end{array}$$

Here the first constraint encodes that each advertiser can derive value from at most one allocation for each page. The second constraint encodes that each advertiser can only derive value from at most n_a slots in these allocations.

The corresponding dual linear program is the following:

$$\begin{array}{rcl} \text{minimize} & \sum_{p,a} z_{p,a} + \sum_a n_a \cdot \beta_a & \text{(Dual)} \\ \forall p, C \in \mathcal{C}_p, a: & z_{p,a} + |C_a| \cdot \beta_a \geq v_p(C, a) & [x_{p,C,a}] \\ \forall p, a: & z_{p,a} \geq 0, \beta_a \geq 0 & \end{array}$$

Initially we set all variables to be zero. As we run PD-Exp for page p, we update primal and dual variables as follows:

— Let allocation $C \in \mathcal{C}_p$ be chosen. We set $x_{p,C,a} = 1$ for each a allocated in C, and set $z_{p,a} = v_p(C, a) - |C_a| \cdot \beta_a$.

— If for some advertiser a, $\sum_{p,C} |C_a| \cdot x_{p,C,a}$ exceeds the capacity limit of n_a, we pick the page p and allocation C with nonzero $x_{p,C,a}$ such that $v_p(C, a)/|C_a|$ is minimized, and simulate the removal of 1 slot by decreasing $x_{p,C,a}$ by $\frac{1}{|C_a|}$. We repeat this until the capacity constraint is respected. Note that because we assumed all slots on the page equally share in the value $v_p(C, a)$, we can decrease the allocation on one page to zero before moving on to the next page. This results in at most one fractionally assigned page per advertiser, yielding a negligible loss.

Now we verify that primal feasibility and dual feasibility are always preserved. The interesting case is to verify that $z_{p,a}$ is always nonnegative. Suppose for contradiction that $z_{p,a}$ is negative for some p, a. Consider the allocation $C \backslash a$ obtained from C by removing assignments for a. By the cross-monotonicity assumption, the total value-shares of other advertisers from C is higher. Therefore $C \backslash a$ has strictly higher discounted value-share, and should be chosen by PD-Exp instead of C, contradiction.

Next, we relate the primal gain to dual gain. We treat advertiser a who is assigned $|C_a|$ slots as if she is assigned $|C_a|$ slots with equal value of $\frac{v_p(C,a)}{|C_a|}$, where this value is at least β_a. Now we can use essentially the same analysis as in the proof of Theorem 4.1 (based on Lemma 2.2) to show that the primal gain is at least ρ_a fraction of the dual gain. Summing over all advertisers and all pages gives us the theorem.

5. SWM WITH ONLINE BIDDERS

As we discussed in Section 3.3, the PA-Indep-Matroid setting is related to SWM with online items. In this section, we show that the PA-Dep problem is related to the following variant of online SWM with online bidders:

SWM with Online Bidders and Item Reassignments. In this variant of online SWM problem, we have offline items and online bidders. 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 we can buy back for free.

In the following, we show that SWM with online bidders can be reduced to the PA-Dep setting. In making the connection, the intended meaning of bidders and items in the context of PA-Dep will be reversed. In particular, items now correspond to offline advertisers, and bidders now correspond to online pages. To prove this reduction, we use the fact that submodular valuation functions admit cross-monotonic value sharing methods [Moulin and Shenker 2001].

LEMMA 5.1. *Given a ρ-competitve algorithm for PA-Dep where every advertiser has capacity 1 and capacity ratio 1, there exists a ρ-competitive algorithm for SWM with online bidders.*

PROOF. Given an instance of SWM with online bidders, we construct a corresponding PA-Dep setting as follows.

Let there be m items numbered $1, \dots, m$. 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 $S \subseteq \{1, \dots, m\}$, include a feasible allocation where for all $j \in S$ slot j is assigned to advertiser j, and all slots outside S are not assigned. Furthermore, the value-share $v_p(C, j)$ of advertiser j is defined as $f(\{1, \dots, j\} \cap S) - f(\{1, \dots, j-1\} \cap S)$.

Clearly the set of feasible allocations defined this way is downward-closed. To verify that the value-sharing function is cross-monotonic, note that if an advertiser is removed from an allocation that corresponds to item set S, the value-share of each advertiser $j \in S$ is $f(\{1, \dots, j\} \cap \{S \backslash a\}) - f(\{1, \dots, j-1\} \cap \{S \backslash a\})$, which is as large as $f(\{1, \dots, j\} \cap S) - f(\{1, \dots, j-1\} \cap S)$ by submodularity. Note also that the value-sharing is defined in a way such that the total value of allocated advertisers in S is equal to $f(S)$.

Note that for this particular PA-Dep instance, every advertiser has a unit capacity, as well as a capacity ratio of 1 as she can win at most one slot in each page.

Now given a ρ-approximation algorithm for PA-Dep, we can simulate it on the above PA-Dep 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-share by doing this is higher than the value-share 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 PA-Dep, we lose a value-share of v in accounting for advertiser j, while in the online SWM problem, by submodularity, bidder b loses a value of at most v. In either case, the gain for the current page in the PA-Dep instance is equal to the gain for the current bidder in the SWM instance. It follows that at the end of process, the algorithm for online SWM performs as well as the algorithm for PA-Dep. Since both problem settings share the same optimal value, our lemma follows. □

By Theorem 4.1, PD-Exp gives a $\frac{1}{2}$-approximation for PA-Dep when both capacity and capacity ratio are one. It follows that we have a $\frac{1}{2}$-approximation algorithm for SWM with online bidders. Furthermore, under the following assumption, PD-Exp 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, a multi-set subset of items becomes available, and the arriving bidder reports a monotone submodular valuation function w.r.t. items in this subset. For each item, the capacity ratio corresponds to the ratio of the number of units of this item to the maximum number of units of this item that is available to a bidder. For this setting, we can apply our result for PA-Dep to get $(1 - \frac{1}{e} - o(1))$-approximation assuming that the minimum capacity ratio of an item is not small.

Efficient implementation of PD-Exp using demand oracles. As we noted in Section 4, the PD-Exp algorithm runs in polynomial time only if we can enumerate all possible configurations on a page, e.g., in the case that the number of ads per page is a constant. Here, we observe that using a *demand oracle*[7] access to submodular valuation functions, one can implement the PD-Exp algorithm for the SWM problem with online bidders in polynomial time even if the number of items per bidder is not a constant. To see this, note that at each step of the PD-Exp algorithm, we need to find a configuration S maximizing $\sum_{a \in S}(v_a(S) - |S_a|\beta_a) = f(S) - \sum_{a \in S} |S_a|\beta_a$ which can be done using a demand oracle access to the submodular valuation function. In updating β_a variables while running the algorithm, we also need to compute the value shares for each advertiser, and thus we need also to have a value oracle access to submodular valuation functions. However, we know that value oracles can be simulated in polynomial time using demand query oracles [Blumrosen and Nisan 2009]. Therefore, having a demand oracle access to valuation functions is sufficient for implementing the PD-Exp algorithm in polynomial time.

6. EMPIRICAL EVALUATION

Our investigation into PA-Dep and Submodular Welfare Maximization with online bidders was initially inspired by the very concrete problem of page-based display ad allocation. Besides being theoretically optimal, a key feature of our algorithm is its simplicity and ease of implementation, allowing one to verify whether it also performs well in practice. In this section, we present experimental results, comparing a page-based 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 varies from the twenties to the hundreds. Advertisers specify complex targeting criteria to define the set of eligible impressions (this gives the bipartite graph between impressions and advertisers), and the *edge weights* capture the “targeting quality” of an advertiser for an impression. The specification of all per-page constraints for each advertiser is non-trivial and hard to describe succinctly. Therefore, to aid reproducibility of these experiments, 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, 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.

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:

[7] A demand oracle for a function f answers the following types of queries: Given a price vector $\{p_1, \ldots, p_n\}$ for items, return a set S maximizing $f(S) - \sum_{j \in S} p_j$?

It is difficult to use a slightly different multiplier for each one of a million impressions, all of which must be updated after each allocation is difficult; bucketing these multipliers does not significantly affect algorithm performance. Further, the use of a normalizing coefficient of $\hat{n}_a$ instead of n_a (a difference of a factor of $1 - o(1)$) in the exponentially-weighted average β_a is a technical requirement to deal with very small weight differences; this can be ignored in practice. For the page-based algorithm, we explicitly solve the LP of Section 4 for each page to enforce the exclusion constraints.

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 I 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.

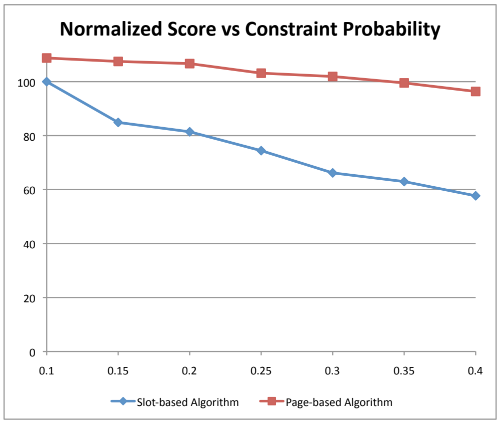

Fig. 2.
Performance vs constraint probability, Publisher B

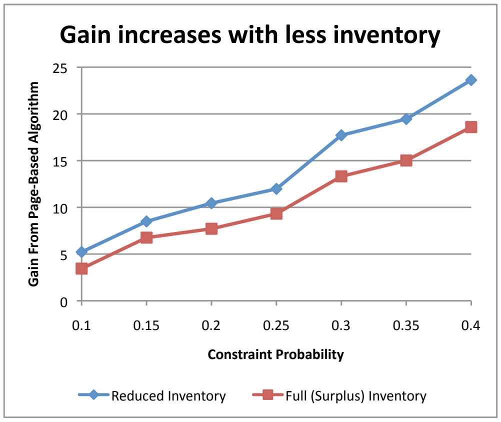

Fig. 3.
Increased gain with reduced inventory, Publisher D

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

Table I. 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.

	Pub A		Pub B		Pub C		Pub D		Pub E		Avg		Gain
Const. 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%

Fig. 2. Performance vs. constraint probability, Publisher D.

few users, it is critically important that early opportunities not be wasted, and page-based algorithms have an even clearer advantage. 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.

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 may vary from one publisher to another, and it is harder to compare these scientifically and publish results of reproducible experiments.

REFERENCES

AGGARWAL, G., FELDMAN, J., MUTHUKRISHNAN, S., AND PÁL, M. 2008. Sponsored search auctions with markovian users. *Internet and Network Economics, 4th International Workshop, WINE 2008. Proceedings.*, 621–628.

AGRAWAL, S., WANG, Z., AND YE, Y. 2009. A dynamic near-optimal algorithm for online linear programming. Working paper posted at http://www.stanford.edu/ yyye/.

ATHEY, S. AND ELLISON, G. 2012. Position auctions with consumer search. To Appear in Quarterly Journal of Economics.

BABAIOFF, M., HARTLINE, J. D., AND KLEINBERG, R. D. 2009. Selling ad campaigns: online algorithms with cancellations. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC-2009), Stanford, California, USA, July 6–10, 2009*. 61–70.

BLUMROSEN, L. AND NISAN, N. 2009. On the computational power of demand queries. *SIAM J. Comput 39,* 4, 1372–1391.

BUCHBINDER, N., JAIN, K., AND NAOR, J. 2007. Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue. In *Proc. ESA*. Springer, 253.

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 PÁL, M. 2009. An online mechanism for ad slot reservations 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 *ACM EC*.

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 *ACM Conference on Electronic Commerce*. 29–38.

FEIGE, U., IMMORLICA, N., MIRROKNI, V. S., AND NAZERZADEH, H. 2008. A combinatorial allocation mechanism with penalties for banner advertising. In *WWW*. 169–178.

FELDMAN, J., HENZINGER, M., KORULA, N., MIRROKNI, V. S., AND STEIN, C. 2010. Online stochastic packing applied to display ad allocation. In *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I*, M. de Berg and U. Meyer, Eds. Lecture Notes in Computer Science Series, vol. 6346. Springer, 182–194.

FELDMAN, J., KORULA, N., MIRROKNI, V. S., MUTHUKRISHNAN, S., AND PÁL, M. 2009a. Online ad assignment with free disposal. In *Internet and Network Economics, 5th International Workshop, WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings*, S. Leonardi, Ed. Lecture Notes in Computer Science Series, vol. 5929. Springer, 374–385.

FELDMAN, J., MEHTA, A., MIRROKNI, V., AND MUTHUKRISHNAN, S. 2009b. Online stochastic matching: Beating 1 - 1/e. In *FOCS*.

HAEUPLER, B., MIRROKNI, V., AND ZADIMOGHADDAM, M. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*.

KARANDE, C., MEHTA, A., AND TRIPATHI, P. 2011. Online bipartite matching with unknown distributions. In *STOC*.

KARP, R. M., VAZIRANI, U. V., AND VAZIRANI, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, B. Awerbuch, Ed. ACM Press, Baltimore, MY, 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 *Internet and Network Economics, 4th International Workshop, WINE 2008. Proceedings.* 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.

LEHMANN, D., LEHMANN, B., AND NISAN, N. 2001. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM Conference on Electronic Commerce (EC-01)*. ACM Press, New York, 18–28.

MAHDIAN, M. AND YAN, Q. 2010. Unpublished Note.

MAHDIAN, M. AND YAN, Q. 2011. Online bipartite matching with random arrivals: A strongly factor revealing lp approach. In *STOC*.

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

MEHTA, A., SABERI, A., VAZIRANI, U. V., AND VAZIRANI, V. V. 2007. Adwords and generalized online matching. *J. ACM 54,* 5.

MENSHADI, H., OVEISGHARAN, S., AND SABERI, A. 2011. Offline optimization for online stochastic matching. In *SODA*.

MIRROKNI, V., GHARAN, S. O., AND ZADIMOGHADDAM, M. 2012. Simultaneous approximations for adversarial and stochastic online budgeted allocation problems. In *SODA*.

MOULIN, H. AND SHENKER, S. 2001. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory 18,* 3, 511–533.

NEMHAUSER, G. L. AND WOLSEY, L. A. 1981. Maximizing submodular set functions: Formulations and analysis of algorithms. In *Studies on Graphs and Discrete Programming*, P. Hansen, Ed. North Holland, Annals of Discrete Mathematics, 11, Amsterdam, 279–301.

OXLEY, J. G. 1992. *Matroid theory*. Oxford University Press.

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

SCHRIJVER, A. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Algorithms and Combinatorics Series, vol. 24. Springer.

TAN, B. AND SRIKANT, R. 2010. Online advertisement, optimization and stochastic networks. *CoRR abs/1009.0870*.

VEE, E., VASSILVITSKII, S., AND SHANMUGASUNDARAM, J. 2010. Optimal online assignment with forecasts. In *ACM EC*.

VONDRAK, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*.

A. PA-INDEP-MATROID AND ONLINE SWM

A.1. Hard v.s. Soft Constraints

We have formulated the allocation constraints as “hard” constraints, in the sense that the slot set allocated to an advertiser a must be an independent set in the matroid of a. We can also formulate a “soft” version of the problem instead. In the soft version, the only allocation constraint is that each slot can go to at most one advertiser. On the other hand, when we account for the value of an advertiser at the end, we require that for each page, the advertiser can only derive value from a slot set that is independent in the matroid.

A helpful fact is that the value of an advertiser is a weighted matroid rank function over the slot set (over all pages) she is allocated. To see this, an advertiser can derive value from a set that obeys the matroid constraint for each page, and a global capacity constraint. Overall, the constraint is given by the intersection of a matroid constraint with a uniform matroid, which is still a matroid. Therefore the soft version of PA-Indep-Matroid is a special case of SWM with online items.

The soft version and hard version of PA-Indep-Matroid are equivalent, in the following sense:

LEMMA A.1. *An algorithm for the hard version is also automatically an algorithm for the soft version with the same approximation ratio. Conversely, given an ρ-approximation algorithm for the soft version, there is a ρ-approximation algorithm for the hard version.*

PROOF. Note that both problems share the same offline optimal objective value. The first claim is obvious, and we only verify the second claim here.

Given a ρ-approximation algorithm $\mathcal{A}$ for the soft version, consider the algorithm $\mathcal{A}'$ for the hard version which works as follows. $\mathcal{A}'$ first simulates $\mathcal{A}$ for each page p, and for each advertiser a, let S_a be the set of slots assigned to a. For every advertiser a, find the weight maximizing subset S'_a of S_a that is independent in the matroid, and assign slots in S'_a to a.

It remains to argue that the final objective value of $\mathcal{A}'$ equals to that of $\mathcal{A}$. Note that in $\mathcal{A}$, a possibly infeasible slot set for a is given for each page, and at the end we choose feasible subsets of these sets to maximize weight subject to an additional capacity constraint. On the other hand in $\mathcal{A}'$, we make sure the slot set for a for each page is feasible and maximizing for the page first, and the rest process is the same. By property of matroids, the slots removed this way would never contribute to the final objective value, and therefore the two processes end up with the same final value. □

Now Lemma 3.3 follows from Lemma A.1 and the fact that the soft version of PA-Indep-Matroid is a special case of SWM with online items.

A.1.1. Slot-Based Algorithms. As the soft version of the problem is a special case of online SWM, whereas algorithms for online SWM are essentially "slot-based" instead of "page-based". It would be interesting to study whether slot-based algorithm can match with the performance of page-based algorithms for the soft version of PA-Indep-Matroid.

For the hard version of the problem, in general, it is easy to see that slot-based algorithms cannot guarantee even a constant factor approximation. However for the special unweighted case where all weights are either zero and one, slot-based algorithm can indeed perform as well as the best page-based algorithm for PA-Indep-Matroid. Mahdian and Yan [Mahdian and Yan 2010] observed that the RANKING algorithm of Karp, Vazirani and Vazirani can be generalized to PA-Indep-Matroid in a natural way, achieving a $1-\frac{1}{e}$ approximation. The generalized RANKING algorithm would never allocate a set of slots to an advertiser that does not satisfy matroid constraint. Therefore for the unweighted case, slot-based algorithm performs as well as page-based algorithms.