Online Ad Assignment with Free Disposal

Jon Feldman[1], Nitish Korula[2], Vahab Mirrokni[1], S. Muthukrishnan[1],
and Martin Pál[1]

[1] Google Inc., 76 9th Avenue, New York, NY 10011
{jonfeld,mirrokni,muthu,mpal}@google.com
[2] Dept. of Computer Science, University of Illinois, Urbana, IL 61801
Work done while at Google Inc., NY
nkorula2@illinois.edu

Abstract. We study an online weighted assignment problem with a set of fixed nodes corresponding to advertisers and online arrival of nodes corresponding to ad impressions. Advertiser a has a contract for $n(a)$ impressions, and each impression has a set of weighted edges to advertisers. The problem is to assign the impressions online so that while each advertiser a gets $n(a)$ impressions, the total weight of edges assigned is maximized.

Our insight is that ad impressions allow for *free disposal*, that is, advertisers are indifferent to, or prefer being assigned more than $n(a)$ impressions without changing the contract terms. This means that the value of an assignment *only* includes the $n(a)$ highest-weighted items assigned to each node a. With free disposal, we provide an algorithm for this problem that achieves a competitive ratio of $1 - 1/e$ against the offline optimum, and show that this is the best possible ratio. We use a primal/dual framework to derive our results, applying a novel exponentially-weighted dual update rule. Furthermore, our algorithm can be applied to a general set of assignment problems including the *ad words* problem as a special case, matching the previously known $1 - 1/e$ competitive ratio.

1 Introduction

Motivation: Display Ads Allocation. Many web publishers (e.g., news sites) have multiple pages (sports, arts, real estate, etc) where they show image, video or text ads. When a visitor to such a web site is exposed to an ad, this is called an "impression." Advertisers typically buy blocks of impressions ahead of time via contracts, choosing blocks carefully to target a particular market segment, typically as part of a more general advertising campaign across various web sites and other media outlets. Once the contract is agreed upon, the advertiser expects a particular number of impressions to be delivered by the publisher over an agreed-upon time period.

The publisher enters all such impression contracts into an ad delivery system. Such systems are typically provided as a service by third party companies, but sophisticated publishers may develop their own software. When a user views one of the pages with ad slots, this system determines the set of eligible ads for that

S. Leonardi (Ed.): WINE 2009, LNCS 5929, pp. 374–385, 2009.

slot, and selects an ad to be shown, all in real time. Because traffic to the site is not known beforehand, it must solve an online matching problem to satisfy the impression contracts. However, before committing to a set of contracts, it would have been already determined using traffic forecasts that the contracts are likely to be fulfillable. Thus, if this were purely a cardinality matching problem, it would typically be easy to solve; what makes the problem challenging is the fact that not all impressions are of equal value to an advertiser (e.g., top vs. side slots, sports vs. arts pages). The publisher is interested not only in filling the impression contracts, but also delivering well-targeted impressions to its advertisers (as measured, e.g., by click-throughs). Thus the ADS, when deciding which ad to serve, has the *additional* goal of maximizing the overall quality of impressions used to fill the contracts. We formulate and study this online optimization problem.

Online Ad Allocation Problem. We have a set of advertisers A known in advance, together with an integer impression contract $n(a)$ for each advertiser $a \in A$. Each $a \in A$ corresponds to a node in one partition of the bipartite graph we define. The set of impressions I forms the nodes of the other partition and they arrive online. When an impression $i \in I$ arrives, its value $w_{ia} \geq 0$ to each advertiser a becomes known (some of the w_{ia}'s are possibly zero). The value w_{ia} might be a prediction of click-through probability, an estimate of targeting quality, or even the output of a function given by the advertiser; we treat this abstractly for the purposes of this work. The impression i must be assigned immediately to some advertiser $a \in A$.

Let $I^a \subseteq I$ be the set of impressions assigned to a during the run of the algorithm. The goal of the algorithm is to maximize overall advertiser satisfaction, i.e., $\sum_{a \in A} S(a, I^a)$ for some satisfaction function S. To encode the impression contracts $n(a)$ as part of S, one possible choice is to say $S(a, I^a) = \sum_{i \in I^a} w_{ia}$ if $|I^a| \leq n(a)$ (and $S(a, I^a) = -\infty$ otherwise). In other words, maximize overall quality without exceeding any of the contracts $n(a)$. As stated, no bounded competitive ratio can be obtained for this problem: just consider the simple case of a single advertiser, $n(a) = 1$, and two items arriving. The first item that arrives has value 100. If it is assigned, then the next item has value 10000; if it is not assigned, the next item has value 1. (In both cases the algorithm achieves less than 1/100th the value of the optimal solution.)

The main insight that inspires our model is that the strict enforcement of the impression contract as an upper bound is inappropriate, since impressions exhibit what is known as the property of *free disposal* in Economics. That is, in the presence of a contract for $n(a)$ impressions, the advertiser is only pleased — or is at least indifferent to — getting *more* than $n(a)$ impressions. Therefore, a more appropriate formulation of the problem is the following. We let I^a_k be the k impressions $i \in I^a$ with the largest w_{ia}. Then, define

$$S(a, I^a) = \sum_{i \in I^a_{n(a)}} w_{ia}.$$

In other words, each advertiser draws its value from its top $n(a)$ impressions, and draws zero value from its remaining impressions (yielding free disposal).

We call this the *display ads* (DA) problem. Free disposal makes the problem tractable; e.g., for the counterexample above with a single advertiser a, the trivial algorithm that assigns all the impressions to that advertiser is optimal. (The general problem with multiple advertisers is, of course, nontrivial.) This choice of S also allows us to tradeoff between quality and contract fulfillment by adding a constant W to each w_{ia}; for large W the problem becomes closer to a pure maximum-cardinality matching.

Our Results and Techniques. Our main technical contribution is an online algorithm for the DA problem with competitive ratio of $1-1/e$, as long as $n(a) \to \infty$. Further, this is the best possible for any (even randomized) online algorithm.

We generalize our algorithm to the case of non-uniform item sizes, the so-called *Generalized Assignment Problem* (GAP). More specifically, we can add "sizes" s_{ia} to the model, where the contract then refers to the total *size* of impressions assigned to an advertiser (and the function S is defined appropriately; see Section 3 for more details). This generalization captures both the DA problem as well as the well-studied *ad words* (AW) problem [19], where the advertisers express budgets B_a (simply set $s_{ia} = w_{ia}$). Our bound of $1 - 1/e$ when sizes are "small" matches the best known ratio for the AW problem. Furthermore, GAP is a unifying generalization that can handle hybrid instances where some advertisers are budget-constrained, and some are inventory constrained.

Our algorithm for the DA problem is inspired by the techniques developed for the online Ad Words (AW) allocation problem in [19], as well as the general primal-dual framework for online allocation problems [4]. The key element of this technique is to develop a *dual update rule* that will maintain dual feasibility as well as a good bound on the gap between the primal and dual solutions. Previous algorithms for related online packing problems such as AW [5] typically update dual (covering) variables by multiplying them by a small factor (such as $1+1/n$) at each step, and adding a term proportional to the increase in primal value. By contrast, our update rule sets the dual variable for each advertiser a to be a carefully weighted average of the weights of the top $n(a)$ impressions currently assigned to a. In fact, the value of the dual variable for an advertiser with a set of impressions I^a is the same as it would be if we *re-ordered the impressions* in increasing order of weight and used the update rules of previous algorithms (as in [4]) on I^a. By choosing our exponentially-weighted update rule, we balance the primal and dual objectives effectively and obtain an optimal algorithm for the DA problem.

Related Work. The related AW problem discussed above is NP-Hard in the offline setting, and several approximations have been designed [6,22,2]. For the online setting, it is typically assumed that every weight is very small compared to the corresponding budget, in which case there exist $(1 - 1/e)$-factor online algorithms [19,4,15,1], and this factor is tight. In order to go beyond the competitive ratio of $1 - \frac{1}{e}$ in the adversarial model, stochastic online variants of the problem have been studied, such as the random order and i.i.d models [15]. In particular, for any ε, a primal-dual $1 - \varepsilon$-approximation has been developed for this problem in the random order model with the assumption that *opt* is larger than $O(\frac{n^2}{\varepsilon^3})$ times each bid [9]. Moreover, a 0.67-competitive algorithm has been recently developed for the (unweighted) max-cardinality version of this problem in the i.i.d. model (without any extra assumption) [12]. Previously, a randomized $(1-\frac{1}{e})$-competitive algorithm for the max-cardinality problem was known in the adversarial model [16]. The online maximum weighted b-matching problem *without free disposal* in the random permutation model has also been studied, and a $\frac{1}{8}$-approximation algorithm has been developed for this problem [17].

Prior to the development of the $(1 - \frac{1}{e})$-approximation algorithm for the offline GAP, various $\frac{1}{2}$-approximation algorithms had been obtained for this problem [8,21,13]. It has been observed that beating the approximation ratio $1 - \frac{1}{e}$ for more general packing constraints is not possible unless NP$\subseteq$ DTIME$(n^{O(\log\log n)})$. However, for GAP with simple knapsack constraints, an improved $1-\frac{1}{e}+\delta$-approximation (with $\delta \approx 10^{-180}$) was developed by Feige and Vondrak [11]. In the online model with small sizes, our approximation factor of $1 - \frac{1}{e}$ is tight.

The offline variants of DA, AW, and GAP are special cases of the problem of maximizing a monotone submodular function subject to a matroid constraint [13]. Recently, the approximation factor for this problem has been improved from $\frac{1}{2}$ to $1 - \frac{1}{e}$ [23], but these algorithms do not work in the online model. The algorithm in [18], although studied for the offline setting, works for the online DA problem and gives a $\frac{1}{2}$-competitive algorithm (discussion below).

2 The Display Ads Problem

In this section, we provide online algorithms for the DA problem with small competitive ratios. Recall that the competitive ratio of an online algorithm for a maximization problem is defined as the minimum, over all possible input sequences, of the ratio between the value obtained by the algorithm and the optimum value on that sequence. We first give a simple upper bound:

Lemma 1. *No deterministic algorithm for the Display Ads problem achieves a competitive ratio better than* $1/2$.

Proof. Consider an instance in which there are two advertisers a_1, a_2 each with capacity 1, and two impressions i_1, i_2. Impression i_1 has value w for both advertisers, and arrives first. Once it has been assigned, i_2 arrives, and has value w for the same advertiser to which i_1 was assigned. Thus we obtain a value of w, while the optimal solution has value $2w$. □

In this section, we show that a greedy algorithm is always 1/2-competitive, matching the bound of Lemma 1. On real instances of the Display Ads problem, though, advertisers request far more than a single impression, and so a natural question is whether one can obtain better deterministic algorithms if $n(a)$ is large for each advertiser a. Also in this section, we answer this question affirmatively, giving an algorithm that achieves a competitive ratio tending to $1 - 1/e$ as $n(a)$ tends to infinity.

The Greedy Algorithm. Consider an algorithm for the DA problem, assigning impressions online. When impression i arrives, what is the benefit of assigning it to advertiser a? This impression can contribute w_{ia} to the value obtained by the algorithm, but if advertiser a already has $n(a)$ impressions assigned to it, one of these impressions cannot be counted towards the value. Let $v(a)$ denote the value of the least valuable impression currently assigned to a (if there are fewer than $n(a)$ such impressions, $v(a) = 0$). Clearly, if $w_{ia} \leq v(a)$, there is no benefit to assigning impression i to advertiser a. Let $A_i = \{a \colon w_{ia} > v(a)\}$; any algorithm should only assign i to an impression in A_i.

Perhaps the simplest algorithm is to assign an impression i to the advertiser $a \in A_i$ that maximizes w_{ia}. The competitive ratio of this naive algorithm is arbitrarily bad: Consider a set of advertisers $\{a^*, a_1, a_2, \dots a_n\}$ each with capacity 1, and impressions $\{i_1, i_2, \dots i_n\}$ that appear in that order. Impression i_j has value $1 + j\varepsilon$ for a^*, and value 1 for a_j. The algorithm above obtains value $1 + n\varepsilon$, while the optimal solution has value $n + n\varepsilon$.

One can do better by noticing that the increase in value by assigning impression i to a is $w_{ia} - v(a)$, and therefore greedily assigning i to the advertiser a maximizing this quantity, which we call the *marginal gain* from assigning i to a.

The following theorem shows that the greedy algorithm (maximizing the marginal gain at each step) is 1/2-competitive:

Theorem 1. *The greedy algorithm is $\frac{1}{2}$-competitive for display ad allocation.*

This theorem is a special case of Theorem 8 in [18] which studies combinatorial allocation problems with submodular valuation functions. This follows from the fact that the valuation function of each advertiser in the online DA problem is submodular in terms of the set of impressions assigned to it, i.e., $\sum_{i \in I^a_{n(a)}} w_{ia}$ is submodular in I^a. Though [18] studied this problem in the offline setting, their greedy algorithm can be implemented as an online algorithm. Other offline $\frac{1}{2}$ and $1 - \frac{1}{e}$-approximation algorithms for a more general problem of submodular maximization under matroid constraints are known [13,23], but these offline algorithms do not provide an online solution.

When $n(a)$ is large for each advertiser a, the upper bound of Lemma 1 does not hold; it is possible to achieve competitive ratios better than 1/2. However, even in this setting, the performance of the greedy algorithm does *not* improve.

Lemma 2. *The competitive ratio of the greedy algorithm is* $1/2$ *even when* $n(a)$ *is large for each advertiser* $a \in A$.

Proof. Let each of advertisers a_1, a_2 have capacity n; suppose there are n copies of impression i_1 with value w to a_1 and $w - 1/n$ to a_2. The greedy algorithm assigns all of these impressions to a_1, obtaining value wn. Subsequently, n copies of impression i_2 arrive, with value w to a_1 and 0 to a_2. Thus, the optimal solution has value $2nw - 1$, while the greedy algorithm only obtains a value of nw. □

The greedy algorithm does badly on the instance in Lemma 2 because it does not take the capacity constraints into account when assigning impressions.

Primal-Dual algorithms for the DA problem. We write a linear program where for each we have variables x_{ia} to denote whether impression i is one of the $n(a)$ most valuable impressions assigned to advertiser a.

$$\textbf{Primal:} \quad \max \sum_{i,a} w_{ia} x_{ia}$$

$$\begin{array}{rcll} \sum_{a} x_{ia} & \leq & 1 & (\forall\ i) \\ \sum_{i} x_{ia} & \leq & n(a) & (\forall\ a) \end{array}$$

$$\textbf{Dual:} \quad \min \sum_{a} n(a)\beta_a + \sum_{i} z_i$$

$$\begin{array}{rcl} \beta_a + z_i & \geq & w_{ia} (\forall i, a) \\ {[x_{ia}, \beta_a, z_i} & \geq & 0] \end{array}$$

The algorithms we consider simultaneously construct feasible solutions to the primal and dual LPs, using the following outline:

- Initialize the dual variables β_a to 0 for each advertiser.
- Subsequently, when an impression i arrives online, assign i to the advertiser $a' \in A$ that maximizes $w_{ia} - \beta_a$. (If this value is negative for each a, leave impression i unassigned.)
- Set $x_{ia'} = 1$. If a' previously had $n(a')$ impressions assigned, let i' be the least valuable of these; set $x_{i'a'} = 0$.
- In the dual solution, set $z_i = w_{ia'} - \beta_{a'}$ and increase $\beta_{a'}$ using an appropriate *update rule* (see below); different update rules give rise to different algorithms/assignments.

The outline above results in a valid integral assignment (primal solution) and a feasible dual solution; to completely describe such an algorithm, we only need to specify the update rule used. We consider the following update rules:

1. **Greedy:** For each advertiser a, β_a is the weight of the lightest impression among the $n(a)$ heaviest impressions currently assigned to a. That is, β_a is the weight of the impression which will be discarded if a receives a new high-value impression.
2. **Uniform Weighting:** For each advertiser a, β_a is the average weight of the $n(a)$ most valuable impressions currently assigned to a. If a has fewer than $n(a)$ assigned impressions, β_a is the ratio between the total weight of assigned impressions and $n(a)$.
3. **Exponential Weighting:** For each advertiser a, β_a is an "exponentially weighted average" (see Def. 1) of the $n(a)$ most valuable impressions.

It is easy to see that the Greedy rule simply gives rise to the greedy algorithm that assigns each impression to the advertiser that maximizes marginal gain. Using Uniform Weighting, one can obtain an improved ratio $\approx 3/4$ on the instance of Lemma 2, as the first n copies of impression i_1 are split evenly between advertisers a_1 and a_2, and thus half the copies of impression i_2 can be assigned to a_1. We state and analyze the Exponential Weighting rule in more detail below, but as a warm-up, we use the primal-dual technique to show that the Uniform Weighting rule gives a 1/2-competitive algorithm.

Lemma 3. *The primal-dual algorithm with Uniform Weighting is* $\frac{1}{2}$*-competitive.*

Proof. We show that the value of the feasible dual solution constructed by the algorithm is at most twice the value of the assignment; by weak duality, this implies that the algorithm is 1/2-competitive. It suffices to show that in any step, the increase in value of the assignment is at least 1/2 of the increase in value of the dual solution. If impression i is assigned to advertiser a, let v be the value of the least valuable impression among the best $n(a)$ impressions previously assigned to a. Thus, the increase in value of the assignment is $w_{ia} - v$. We set $z_i = w_{ia} - \beta_a \leq w_{ia} - v$, as the least valuable impression is worth no more than the average. The increase in β_a is precisely $\frac{1}{n}(w_{ia} - v)$, and hence the total increase in the dual objective function is at most $2(w_{ia} - v)$. □

Using the Greedy Rule, β_a is simply the weight of the edge/impression that will be discarded, while with Uniform Weighting, β_a is the average of all the best $n(a)$ weights currently assigned to a. The disadvantage of the first approach is that it only takes into account the *least* valuable impression, ignoring how much capacity is unused. For Uniform Weighting, Lemma 3 showed that the increase in dual value is $(w_{ia} - v) + (w_{ia} - \beta_a)$, but as one can only use the fact that $v \leq \beta_a$, we get a ratio of 2. To obtain a $(1 - 1/e)$-competitive algorithm, we use an intermediate exponentially-weighted average in which the less valuable impressions are weighted more than the more valuable ones, as follows:

Definition 1 (Exponential Weighting). *Let* $w_1, w_2, \ldots w_{n(a)}$ *be the weights of impressions currently assigned to advertiser a, sorted in non-increasing order. Let* $\beta_a = \frac{1}{n(a)\cdot\left((1+1/n(a))^{n(a)}-1\right)} \sum_{j=1}^{n(a)} w_j \left(1 + \frac{1}{n(a)}\right)^{j-1}$.

Theorem 2. *The primal-dual algorithm with the Exponential Weighting update rule has a competitive ratio of* $(1 - 1/e)$ *as* $n(a) \rightarrow \infty$ *for each advertiser a.*

Proof. Let $e_n = (1 + 1/n)^n$; we have $\lim_{n\rightarrow\infty} e_n = e$. Analogous to the proof of Lemma 3, it suffices to show that at each impression/step of the algorithm, the increase in the value of the assignment is at least $(1 - 1/e_{n(a)})$ times the increase in value of the feasible dual solution, where a is the advertiser to which this impression is assigned.

As before, let impression i be assigned to advertiser a, and let v be the value of the least valuable impression among the best $n(a)$ impressions previously assigned to a. Thus, the increase in value of the assignment is $w_{ia} - v$, and we set $z_i = w_{ia} - \beta_a$. It remains to bound the increase in β_a, which we do as follows.

Let β_o, β_n denote the old and new values of β_a respectively. Suppose that after i is assigned to a, it becomes the most valuable impression assigned to a. Then, we have $\beta_n = (1+1/n)\beta_o - \frac{ve_n}{n(e_n-1)} + \frac{w_{ia}}{n(e_n-1)}$. Thus, $n(\beta_n - \beta_o) = \beta_o - \frac{ve_n}{e_n-1} + \frac{w_{ia}}{e_n-1}$. Therefore, the total dual increase, which is the sum of z_i and n times the increase in β_a is $(w_{ia} - \beta_o) + \beta_o - \frac{ve_n}{e_n-1} + \frac{w_{ia}}{e_n-1} = \frac{(w_{ia}-v)e_n}{e_n-1}$. Therefore, the ratio between the increase in assignment value and dual objective function is $1 - 1/e_n$.

We assumed above that i became the most valuable impression assigned to a; what if this is not true? It is not difficult to verify that in this case, the increase in β_a is *less* than otherwise; to see this, note that if it is the jth most valuable impression, the contribution of w_{ia} to β_a must be multiplied by a factor of $(1 + 1/n)^{j-1}$ compared to the previous case, but the contributions of $j - 1$ more valuable impressions will be decreased by a factor of $(1 + 1/n)$. □

Theorem 3 ([19]). *No algorithm achieves a competitive ratio of greater than $1 - 1/e$ for the display ad allocation problem. This is true even with weights in $\{0, 1\}$, and for randomized algorithms against oblivious adversaries.*

The lower bound of Theorem 3 was proved by [19] for the *Ad words* problem; the example they give is a valid instance of the Display Ads problem, and hence the same lower bound applies. Thus, our primal-dual algorithm with the Exponential Weighting update rule is optimal for the DA problem.

3 The Generalized Assignment Problem

In the Generalized Assignment Problem (GAP), a set A of bins/machines and a set I of items/jobs is given. Each bin $a \in A$ has a capacity C_a; for each item i and bin a, we have a size s_{ia} that item i occupies in bin a and a weight/profit w_{ia} obtained from placing i in a. (Alternately, one can think of GAP as a scheduling problem with s_{ia} as the processing time job i takes on machine a, and with w_{ia} being the value gained from scheduling job i on machine a.) Note that the special case of GAP with a single bin/machine is simply the Knapsack problem.

We first note that GAP captures both the Display Ads problem and the Ad Words problem as special cases, where bins correspond to advertisers and items to impressions. The DA problem is simply the special case in which $s_{ia} = 1$ for all i, a, and the AW problem is the special case in which $w_{ia} = s_{ia}$ for all i, a.

For the offline GAP, the best approximation ratio known is $1 - 1/e + \delta$, where $\delta \approx 10^{-180}$ [11]; this improves on the previous $(1-1/e)$-approximation of [14]. In an online instance of GAP, the set of bins A is known in advance, together with the capacity of each bin. Items arrive online, and when item i arrives, w_{ia} and s_{ia} are revealed for each $a \in A$. The only previous work on online GAP appears to have been for the special case corresponding to the Knapsack problem [3].

Recall that without free disposal, the online Display Ads problem was intractable. We make a similar assumption to solve GAP online; here, we assume that we can assign items of total size more than C_a to bin a, but that the total value derived by bin a is given by the most profitable set of assigned items that actually fits within capacity C_a. (Note that such an assumption is not necessary for the easier Ad Words problem, in which the value/weight of an item in a bin is equal to its size; thus, there is never a need for over-assignment.) Thus, an online algorithm for GAP immediately gives algorithms with the same competitive ratio for the DA and AW problems. In fact, an algorithm for GAP allows one to simultaneously handle ad allocation problems in which some bidders have budget constraints and others have inventory constraints. Unfortunately, we have:

Lemma 4. *No deterministic online algorithm for GAP with free disposal can achieve a competitive ratio better than* $n^{-1/2}$.

Given this lower bound, for the rest of this section, we consider the case of *small items*; that is, we assume that for each item i and bin a such that $w_{ia} > 0$, $s_{ia} \leq \varepsilon C_a$.[1] This is a reasonable assumption for both the DA and AW problems, where contracts are for large numbers of impressions or individual bids are small compared to budgets. We refer to GAP restricted to such instances – where no individual item can occupy more than an ε fraction of any bin – as ε-GAP. Let $e_{1/\varepsilon} = (1+\varepsilon)^{1/\varepsilon}$; we prove the following theorem:

Theorem 4. *There is a* $(1-1/e)$*-competitive algorithm for* ε*-GAP as* $\varepsilon \to 0$.[2]

Proof Sketch. We construct a feasible dual solution (primal and dual linear programs for GAP are given below) as in the proof of Theorem 2, but a problem arises in dealing with non-uniform sizes. It may sometimes be necessary for the algorithm to place an item in a bin even when doing so would *decrease* the value of the solution; this holds even when item sizes are all less than ε times the bin capacities. The intuition is as follows: Suppose an item i arrives with value/size ratio significantly better than the average for a given bin a; it is clear that we should take it, and discard the existing items. (The inability to do this provides the lower bound of Lemma 4.) But if the items already in the bin are larger than the new item, one may lose value by discarding the existing items. This difficulty appears because in integral solutions an item cannot continuously move from being in the bin to outside. We deal with this issue by having the algorithm act as though it *could* derive value from such fractional solutions, in which the item of lowest value/size ratio is partly in the bin, and the value obtained from this item depends on how much of it is in the bin. Under this metric, we show the algorithm's (fractional) value is at least $(1-1/e_{1/\varepsilon})$ times that of a feasible dual solution. Since the algorithm does not truly obtain any integral value from such partially assigned items, it loses at most the value of these items, which is an ε fraction of its overall value. Thus, we obtain an integral solution which achieves an approximation ratio of $(1-1/e_{1/\varepsilon})(1-\varepsilon)$.

$$\textbf{Primal:} \quad \max \sum_{i,a} w_{ia} x_{ia}$$
$$\sum_a x_{ia} \leq 1 \qquad (\forall\, i)$$
$$\sum_i s_{ia} x_{ia} \leq C_a \qquad (\forall\, a)$$

$$\textbf{Dual}: \quad \min \sum_a C_a \beta_a + \sum_i z_i$$
$$s_{ia}\beta_a + z_i \geq w_{ia} (\forall i, a)$$
$$[x_{ia}, \beta_a, z_i \geq 0]$$

[1] This lower bound does not apply to *randomized* algorithms; see Section 4.

[2] More formally, we obtain a ratio of $(1-1/e_{1/\varepsilon})(1-\varepsilon)$ for ε-GAP. This is greater than 1/2 for $\varepsilon \leq 0.17$.

4 Extensions and Future Work

Randomized Algorithms and Lower Bounds. For the basic Display Ads problem, we showed an upper bound of $(1 - 1/e)$ on the competitive ratio of all algorithms, and a deterministic algorithm that matches this bound when $n(a)$ is large. Further, Lemma 1 shows that no deterministic algorithm has competitive ratio larger than $1/2$ when $n(a)$ is small; does this bound also apply to randomized algorithms? The randomized algorithm of [16] gets a competitive ratio of $1-1/e$ for the unweighted case. Extending this result to the weighted case seems difficult; a new approach may be necessary.

Similarly, Lemma 4 shows that no deterministic online algorithm for GAP has a competitive ratio better than $n^{-1/2}$. One can avoid this bound using randomization: Toss a coin to determine whether bins should accept only *large* items (that occupy more than 1/3 the bin), or only small items (that occupy at most 1/3 the bin.) In the latter case, use the algorithm of Theorem 4; in the former case, have each bin accept a single item. Since each bin can accept only two big items, we obtain a constant-competitive algorithm in both cases. (A similar observation was also made in [3] for the easier Knapsack problem.) Optimizing constants, we obtain the following theorem:

Theorem 5. *There is a* 0.15*-competitive randomized online algorithm for GAP.*

Extending these results for the online GAP to more general packing problems is an interesting subject of study. In particular, this idea may be applicable to packing problems with *sparse* constraint matrices; see [20,7] for recent work on the offline versions of these problems.

General non-linear valuation functions. The display ad business is performed through a set of pre-determined contracts. Hence, in many settings, the *number* of impressions assigned to an advertiser is an important quality measure in addition to the total valuation (or total weight) of the impressions. In other words, the valuation (or utility) of an advertiser a for receiving a set I^a of impressions is $v_a(I^a) = \sum_{i \in I^a_{n(a)}} w_{ia} + f_a(|I^a|)$ where $f_a : N \to N$ is a non-decreasing function of the number of impressions assigned to a. We may also assume that $f_a(x) = f_a(n(a))$ for any $x \geq n(a)$. The corresponding online ad allocation problem here is to assign impressions to advertisers and maximize $\sum_{a \in A} v_a(I^a)$.

Depending on various quality measures, this function f_a could be concave or convex. A convex function f_a models the guaranteed delivery property of advertisers in that receiving a number of impression close to $n(a)$ is very important. A concave function f_a, on the other hand, captures the diminishing return property of extra impressions for advertisers. We observe that for convex functions f, the ad allocation problem becomes inapproximable, even in the offline case; this hardness result uses a reduction from a banner ad allocation problem with penalties studied in [10]. On the other hand, if all functions f_a are concave, the problem becomes a special case of submodular valuation and the greedy algorithm gives a $\frac{1}{2}$-competitive algorithm. An interesting question is whether the competitive ratio of $\frac{1}{2}$ can be improved to $1 - \frac{1}{e}$.

"Underbidding" and Incentives. One disadvantage of using the free disposal property is that it may incentivize advertisers to declare smaller $n(a)$, in the hope of getting more impressions in the final allocation. We can partially address this concern by modifying the algorithm slightly so that the sum of weights of *all* impressions assigned to a is at most twice the sum of weights of the top $n(a)$ impressions:

Theorem 6. *There is a* $\frac{1-1/e}{2}$*-competitive algorithm for the DA problem such that for each advertiser,* $\sum_{i \in I^a} w_{ia} \leq 2 \sum_{i \in I^a_{n(a)}} w_{ia}$.

To prove this theorem, one simply needs to use the Exponential Weighting update rule but double β_a for each a; we omit details from this extended abstract.

Concluding Remarks: We have used free disposal to solve the online DA problem with a competitive ratio of $1 - 1/e$. An outstanding issue is to understand how free disposal affects the incentives of advertisers, who may be led to speculate. (Note that even the sub-optimal algorithm of Theorem 6 only bounds the total weight of impressions assigned to an advertiser, not the number of impressions received.) A model for incentives must simultaneously handle contract selection/pricing and the online ad allocation problem; this is an interesting subject of future research.

References

1. Alaei, S., Malekian, A.: Maximizing sequence-submodular functions (2009) (manuscript)
2. Azar, Y., Birnbaum, B., Karlin, A.R., Mathieu, C., Nguyen, C.T.: Improved Approximation Algorithms for Budgeted Allocations. In: Proc. Automata, Languages and Programming (2008)
3. Babaioff, M., Hartline, J., Kleinberg, R.: Selling ad campaigns: Online algorithms with cancellations. In: ACM EC (2009)
4. Buchbinder, N., Jain, K., Naor, J.S.: Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 253–264. Springer, Heidelberg (2007)
5. Buchbinder, N., Naor, J.: The Design of Competitive Online Algorithms via a Primal-Dual Approach. Foundations and Trends in Theoretical Computer Science 3(2-3), 93–263 (2007)
6. Chakrabarty, D., Goel, G.: On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and GAP. In: Proc. FOCS, pp. 687–696 (2008)
7. Chekuri, C., Ene, A., Korula, N.: Unsplittable flow in paths and trees and column-restricted packing integer programs. In: Dinur, I., et al. (eds.) APPROX and RANDOM 2009. LNCS, vol. 5687, pp. 42–55. Springer, Heidelberg (2009)
8. Chekuri, C., Khanna, S.: A PTAS for the multiple knapsack problem. In: 11th ACM-SIAM Symp. on Discrete Algorithms (SODA), pp. 213–222 (2000)

9. Devanur, N., Hayes, T.: The adwords problem: Online keyword matching with budgeted bidders under random permutations. In: Proceedings of the 10th ACM Conference on Electronic Commerce, pp. 71–78 (2009)
10. Feige, U., Immorlica, N., Mirrokni, V., Nazerzadeh, H.: A combinatorial allocation mechanism for banner advertisement with penalties. In: WWW (2008)
11. Feige, U., Vondrak, J.: Approximation algorithms for allocation problems: Improving the factor of 1-1/e. In: FOCS (2006)
12. Feldman, J., Mehta, A., Mirrokni, V., Muthukrishnan, S.: Online stochastic matching: Beating 1 - 1/e. In: FOCS (to appear, 2009)
13. Fisher, M., Nemhauser, G., Wolsey, L.: An analysis of the approximations for maximizing submodular set functions II. Math. Prog. St. 8, 73–87 (1978)
14. Fleischer, L., Goemans, M., Mirrokni, V.S., Sviridenko, M.: Tight approximation algorithms for maximum general assignment problems. In: Proc. SODA (2006)
15. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. In: SODA, pp. 982–991 (2008)
16. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for online bipartite matching. In: Proc. STOC (1990)
17. Korula, N., Pal, M.: Algorithms for secretary problems on graphs and hypergraphs. In: Albers, S., et al. (eds.) ICALP 2009, Part II. LNCS, vol. 5556, pp. 508–520. Springer, Heidelberg (2009)
18. Lehman, Lehman, Nisan, N.: Combinatorial auctions with decreasing marginal utilities. Games and Economic Behaviour, 270–296 (2006)
19. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. In: FOCS (2005)
20. Pritchard, D.: Approximability of Sparse Integer Programs. In: Proceedings of the 17th Annual European Symposium on Algorithms, pp. 83–94 (2009)
21. Shmoys, D., Tardos, E.: An approximation algorithm for the generalized assignment problem. Mathematical Programming 62(3), 461–474 (1993)
22. Srinivasan, A.: Budgeted Allocations in the Full-Information Setting. In: Goel, A., Jansen, K., Rolim, J.D.P., Rubinfeld, R. (eds.) APPROX and RANDOM 2008. LNCS, vol. 5171, pp. 247–253. Springer, Heidelberg (2008)
23. Vondrak, J.: Optimal approximation for the submodular welfare problem in the value oracle model. In: STOC (2008)