Online Stochastic Packing Applied to Display Ad Allocation

Jon Feldman[1], Monika Henzinger[2], Nitish Korula[3], Vahab S. Mirrokni[1], and Cliff Stein[4]

[1] Google Research, 76 9th Ave, New York, NY 10011, U.S.A.
{jonfeld,mirrokni,cstein}@google.com
[2] University of Vienna, Austria
monika.henzinger@univie.ac.at
[3] University of Illinois at Urbana-Champaign
nkorula2@illinois.edu
[4] Google Research and Columbia University, New York, NY

Abstract. Inspired by online ad allocation, we study online stochastic packing integer programs from theoretical and practical standpoints. We first present a near-optimal online algorithm for a general class of packing integer programs which model various online resource allocation problems including online variants of routing, ad allocations, generalized assignment, and combinatorial auctions. As our main theoretical result, we prove that a simple dual training-based algorithm achieves a $(1-o(1))$-approximation guarantee in the random order stochastic model. This is a significant improvement over logarithmic or constant-factor approximations for the adversarial variants of the same problems (e.g. factor $1-\frac{1}{e}$ for online ad allocation, and $\log(m)$ for online routing). We then focus on the online display ad allocation problem and study the efficiency and fairness of various training-based and online allocation algorithms on data sets collected from real-life display ad allocation system. Our experimental evaluation confirms the effectiveness of training-based algorithms on real data sets, and also indicates an intrinsic trade-off between fairness and efficiency.

1 Introduction

Online stochastic optimization is a central problem in operations research with many applications in dynamic resource allocation. In these settings, given a set of resources, demands for the resources arrive online, with associated values; given a general prior about the demands, one has to decide whether and how to satisfy (i.e., allocate the desired resources to) a demand when it arrives. The goal is to find a valid assignment with maximum total value. Such problems appear in many areas including online routing [7,3], online combinatorial auctions [9], online ad allocation problems [25,10,12], and online dynamic pricing and inventory management problems. For example, in routing problems, we are given a network with capacity constraints over edges; customers arrive online and bid for a subset of edges (typically a path) in the network, and the goal

M. de Berg and U. Meyer (Eds.): ESA 2010, Part I, LNCS 6346, pp. 182–194, 2010.

is to assign paths to new customers so as to maximize the total social welfare. Similarly, in online combinatorial auctions, bidders arrive online and may bid on a subset of resources; the auctioneer should decide whether to sell those resources to the bidder. In the display ads problem, when users visit a website, the website publisher has to choose ads to show them so as to maximize the value of the displayed ads. In this paper, we study these online stochastic resource allocation problems from theoretical and practical standpoints. Our theoretical results apply to a general set of problems including all those discussed above. Our practical results apply to the problem of display ads and give additional validation of our theoretical models and results.

More specifically, we consider the following general class of packing integer programs (PIP): Let J be a set of m *resources*; each resource $j \in J$ has a capacity c_j. The set of resources and their capacities are known in advance. Let I be a set of n *agents* that arrive one by one online, each with a set of *options* O_i. Each option $o \in O_i$ of agent i has an associated *value* $w_{io} \geq 0$ and requires $a_{ioj} \geq 0$ units of each resource j. The set of options and the values w_{io} and a_{ioj} arrive together with agent i. When an agent arrives, the algorithm has to immediately decide whether to assign the agent and if so, which option to choose. The goal is to find a maximum-value allocation that does not allocate more of any resource than is available.

In the *adversarial* or worst-case setting, no online algorithm can achieve any non-trivial competitive ratio; consider the simple case of one resource with capacity one and two agents. For each agent there are just two options, namely to get the resource or not to get it. If an agent gets the resource, he uses its whole capacity. The first agent has value 100 for getting the resource and value 0 for not getting the resource. If he is assigned the resource, then the value of the second agent for getting the resource is 10000, otherwise it is 1. In both cases the algorithm achieves less than 1/100th of the value of the optimal solution. This example can easily be generalized to show that even randomized algorithms cannot achieve non-trivial competitive ratios, even if there is a single resource:

Theorem 1. *There is no $o(\log n / \log\log n)$-approximation for the online PIP.*

Since in the adversarial setting the lack of prior information about the arrival rate of different types of agents implies strong impossibility results, it is natural to consider *stochastic* settings for online allocation problems, where we may have some prior information about the arrival rate of different types of agents. In particular, we consider the *random-order stochastic model*, in which the agents, their options and associated values may be chosen by an adversary, but the order in which agents arrive is random. We present a training-based online algorithm for the general class of packing integer programs described above and prove that in the random-order stochastic model, it achieves an approximation ratio of $1-\varepsilon$, where ε is a function of the parameters of the integer program[1]; more precisely, ε

[1] In this context, an "α-approximation" means that with high probability under the randomness in the stochastic model, the algorithm achieves at least an α fraction of the value (efficiency) of the offline optimal solution for the actual instance.

measures how large a fraction of any resource can be demanded by a single agent, or how much a single agent's value contributes to the total objective. Thus, as agents become infinitesimally small, we obtain nearly optimal solutions. This result also implies the same result in the i.i.d. model[2].

Our *dual-based* algorithm for the stochastic PIP problem observes the first ε fraction of the input and then solves an LP on this instance. (This requires knowing the number of agents in advance, at least approximately; Theorem 1 can be generalized to show that this is unavoidable for any sub-logarithmic approximation.) For each resource, the corresponding dual variable extracted from this LP serves as a *(posted) price per unit* of the resource for the remaining agents. The algorithm allocates to each remaining agent the option maximizing his *utility*, defined as the difference between the value of an option and the price he must pay to obtain the necessary resources. We prove that this algorithm provides a $1 - \varepsilon$ approximation for the large class of natural packing problems we consider, provided that no individual option for any agent consumes too much of any resource or provides too large a fraction of the total value. Specifically we show the following result. Recall that n and m denote the number of agents and resources respectively; q denotes $\max_i |O_i|$ and OPT the value of an optimal off-line solution to the PIP problem.

Theorem 2. *The Dual-Based algorithm is* $(1-O(\varepsilon))$*-competitive for the online stochastic PIP problem with high probability, as long as:*
(1) $\max_{i,o}\left\{\frac{w_{io}}{\mathrm{OPT}}\right\} \leq \frac{\varepsilon}{(m+1)(\ln n+\ln q)}$ *and (2)* $\max_{i,o,j}\left\{\frac{a_{ioj}}{c_j}\right\} \leq \frac{\varepsilon^3}{(m+1)(\ln n+\ln q)}$.

Applications: Theorem 2 has many applications; we elaborate on several, including routing problems, online combinatorial auctions, the display ad problem, and the adword allocation problem. For each of these problems, we improve on the known results for the online version. In each, we will comment on the interpretation of the two conditions of Theorem 2 in that application. Note that in condition (2), one might wish the dependence of ε on the input parameters to be linear; this does not seem possible in general. However, for specific applications, one may be able to exploit the structure of the LP to prove tighter bounds; we omit details from this extended abstract. Our experimental results show that in practice, one may be able to use ε much smaller than required by the theorem; in particular, for the DA problem, we sample only 1% of the input and obtain a competitive ratio of ≈ 0.89.

In the *online routing* problem, we are initially given a network with capacity constraints over the m edges. When a customer $i \in I$ arrives online, she wishes to send d_i units of flow between some vertices s_i and t_i, and derives w_i units of value from sending such flow. Thus, the set of options O_i for customer i is the set of all $s_i - t_i$ paths in the network. The algorithm must pick a set of customers $I^* \subseteq I$, and satisfy their demands by allocating a path to each of them while respecting the capacity constraints on each edge; the goal is to maximize the

[2] In the i.i.d model each agent arrives independently and identically drawn from a fixed but unknown probability distribution over the set of possible types of agents [17]. Our random-order stochastic model captures the i.i.d model.

total value of satisfied customers. For this problem, the dual variables learned from the sample yield a price for each edge; each customer is allocated the minimum-cost $s_i - t_i$ path if its cost is no more than w_i. In road networks, for instance, these dual variables can be interpreted as the tolls to be charged to prevent congestion. Theorem 2 applies when the contributions of individual agents/vehicles to the total objective or to road congestion are small. As one such example, consider congestion pricing for Manhattan: over a million vehicles enter or leave the island daily, and each of the 12 bridges and tunnels has an annual average daily traffic of over 50,000; using extremely conservative estimates of bridge capacity, we obtain $\varepsilon \approx 0.15$. Online routing problems have been studied extensively in the adversarial model when demands can be large, and there are *(poly)-logarithmic* lower and upper bounds even for special cases [3,7]. Our approach gives a $(1 - o(1))$-approximation for the described stochastic variants of these problems when the demands of individual agents are small.

In the *combinatorial auction* problem, we are initally given a set J of m goods, with c_j units for each good $j \in J$. Agents arrive online, and the options for agent i may include different bundles of goods he values differently; option $o \in O_i$ provides w_{io} units of value, and requires a_{ioj} units of good j. We wish to find a valid allocation maximizing social welfare. Here, the dual variables learned from the sample yield a price per unit of each good; each agent picks the option that maximizes his utility. Here Theorem 2 applies as long as no individual agent controls a large fraction of the market, and as long as the set of options for any single agent is at most exponential in the number of resources. These conditions often hold, as in cases when bidders are single-minded or the number of bundles they are interested in is polynomial in n, or if their options correspond to using different subsets of the resources. We also observe that the posted prices result in a take-it-or-leave it auction, and thus a truthful online allocation mechanism. Revenue maximization in online auctions using sequence item pricing has been explored recently in the literature [5,9]. [9] achieves constant-factor approximations for these problems in more general models than we consider.

In the *Display Ads Allocation (DA)* problem [12], there is a set J of m advertisers who have contracted with a web publisher for their ads to be shown to visitors to the website. When a visitor is shown an ad, this is called an "impression". The contract an advertiser j buys specifies an integer upper bound on the number $n(j)$ of impressions that he is willing to pay for. A set I of visitors arrives online; visitor i has value $w_{ij} \geq 0$ for advertiser j. Each visitor can be assigned to at most one advertiser, i.e., there are m options for an impression, and each option o has $a_{ioj} = 1$ for advertiser j. The goal is to maximize the value of all the assigned impressions. The dual variables learned from the sample yield a discount factor β_j for each advertiser j; the algorithm is to assign an impression to advertiser j that maximizes $w_{ij} - \beta_j$. Contracts for advertisers may involve hundreds of thousands of impressions, so the contribution of any one impression/agent is small. As publishers vary greatly in the amount of web traffic they receive, it is relatively meaningless to define a "typical" instance; however, a publisher with 30 million impressions, and 20 advertisers who have each signed contracts for at least 500, 000 impressions, satisfies the hypotheses of Theorem 2 with $\varepsilon \approx 0.089$. The adversarial online DA problem was considered in [12], which showed that the problem was 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 [12], and this is the best possible. For the *unweighted* (max-cardinality) version of this problem in the i.i.d. model, a 0.67-competitive algorithm has been recently developed [13]; this improves the known $1 - \frac{1}{e}$-approximation algorithm for online stochastic matching [19].

The *AdWords (AW)* problem [25,10] is related to the DA problem; here we allocate impressions resulting from search queries. Advertiser j has a budget $b(j)$ instead of a bound $n(j)$ on the number of impressions. Assigning impression i to advertiser j consumes w_{ij} units of j's budget instead of 1 of the $n(j)$ slots, as in the DA problem. Several approximation algorithms have been designed for the *offline* AW problem [8,26,4]. For the online setting, if every weight is very small compared to the corresponding budget, there exist $(1 - \frac{1}{e})$-competitive online algorithms [25,6,18,2], and this factor is tight. In order to go beyond the competitive ratio of $1 - \frac{1}{e}$ in the adversarial model, stochastic online settings have been studied, such as the random order and i.i.d models [18]. Devanur and Hayes [10] described a dual-based $(1 - \varepsilon)$-approximation algorithm for this problem in the random order model, with the assumption that OPT is larger than $O(\frac{m^2}{\varepsilon^3})$ times each w_{ij}, where m is the number of advertisers; Theorem 2 can be viewed as generalizing this result to a much larger class of problems.

Experimental Validation: For the applications described above, stochastic models are reasonable as the algorithm often has an idea of what agents to expect. For example, in the Display Ad Allocation problem, agents correspond to users visiting the website of a publisher who has sold contracts to advertisers. As the publisher most likely sees similar user traffic patterns from day to day, he has an idea of the available ad inventory based on historical data. In Section 3, we perform preliminary experiments on real instances of the DA problem, using actual display ad data for a set of anonymous publishers. As with any real application, there are additional features of the problem, and in the one we considered, both *fairness* and efficiency were important metrics. Hence, we also evaluated our algorithms for fairness (see Section 3 for a brief discussion); we compared our training-based algorithm with algorithms from [12] designed for the adversarial setting, as well as hybrid algorithms combining the two approaches. Our experimental results validate Theorem 2 for this application, as they show that on this real data set, even sampling 1% of the input (i.e. choosing $\varepsilon = 0.01$) suffices to obtain efficiency of $\approx 89\%$, significantly better than the pure online algorithms from [12], which are in turn much better than a simple greedy approach.

Other Related Work: Our proof technique is similar to that of [10] for the AW problem; it is based on their observation that dual variables satisfy the complementary slackness conditions of the first ε fraction of impressions and *approximately* satisfy these conditions on the entire set. A key difference is that in the AW problem, the coefficients for variable x_{ij} in the linear program are the *same* in both the constraint and the objective function. That is, the contribution an impression makes to an advertiser's value is identical to the amount of budget it consumes; in contrast, these coefficients are unrelated in the general class of packing problems that we study. Further, the structure of the LP considered by [10] is highly restricted, with each variable appearing in precisely two constraints. Thus, our paper provides a simple proof of a more general result than [10], and we experimentally validate these results for the DA problem.

The random-order model has been considered for several problems, often called *secretary* problems. Without additional assumptions (as in Theorem 2), constant-competitive (but no better) algorithms can be obtained for problems such as online Knapsack, or finding maximum weight independent sets in classes of matroids. (See [5] for a survey.) Specifically for the DA problem, the results of [21] imply that the random-order model permits a 1/8-competitive algorithm even without using the free disposal property or the conditions of Theorem 2.

There have been recent results regarding ad allocation strategies in display advertising in hybrid settings with both contract-based advertisers and spot market advertisers [15,14]. Our results in this paper may be interpreted as a class of *representative bidding strategies* that can be used on behalf of contract-based advertisers competing with the spot market bidders [15].

Results similar to ours were recently posted in a working paper[1]. The dependence of ε on the input parameters is almost identical; however, the authors of [1] also observe that if one is willing to "reprice" the resources after every ε fraction of the input, the cubic dependence of ε can be reduced to quadratic. Also recently, Vee *et al.* [27] showed that for assignment-type problems (including the DA and AW problems, but not other problems considered in this paper), if one knows the distribution from which inputs are drawn, a similar dual-based algorithm can be used to obtain online allocations that are nearly optimal for certain "well-conditioned" convex objectives. The results from both these papers were obtained independently of ours, and posted publicly subsequent to the submission of an earlier version of this paper, which included our main result.

2 A Dual-Based Algorithm

In this section, we present the dual-based algorithm for the online stochastic packing problem and prove that, under our (practically-motivated) assumptions, it achieves an approximation factor of $1 - \varepsilon$. Recall that there is a set I of "agents"; agent $i \in I$ has a set of mutually exclusive options O_i, and we use an indicator variable x_{io} to denote whether agent i selects alternative $o \in O_i$. Each option for an agent may have a different "size" in each constraint; we use a_{ioj} to denote the size in constraint j of option o for agent i.

Recall that w_{io} is the value from selecting option o for agent i, and c_j is the "capacity" of constraint j. That is, our goal is to maximize $w^T x$ while picking at most one option for each agent, and subject to $Ax \leq c$. Subsequently, we normalize A, c such that c is the all-1's vector, and write the (normalized) primal linear program below. We also use the dual linear program, which introduces a variable β_j for each constraint j.

Primal-LP

$$\max \sum_i \sum_{o \in O_i} w_{io} x_{io}$$

$$\begin{aligned} \sum_{o \in O_i} x_{io} &\leq 1 & (\forall\, i) \\ \sum_{i,o} a_{ioj} x_{io} &\leq 1 & (\forall\, j) \\ x_{io} &\geq 0 & (\forall\, i, o) \end{aligned}$$

Dual-LP

$$\min \sum_j \beta_j + \sum_i z_i$$

$$\begin{aligned} z_i + \sum_j \beta_j a_{ioj} &\geq w_{io} & (\forall i, o) \\ \beta_j, z_i &\geq 0 & (\forall i, j) \end{aligned}$$

Let n be the total number of agents, $q = \max_i |O_i|$ be the maximum number of options for any agent, and m be the number of constraints. We say that the *gain* from option $o \in O_i$ is $w_{io} - \sum_j \beta_j^* a_{ioj}$. The Dual-Based Algorithm proceeds as follows:

1. Let S denote the first εn agents in the sequence. For the purposes of analysis, these agents are not selected. (Our implementations may assign these agents according to some online algorithm.)
2. Solve the **Dual-LP** on the agents in S, with the objective function containing the term $\varepsilon \beta_j$ instead of β_j for each $j \in [m]$. (This is equivalent to reducing the capacity of a constraint from 1 to ε; we refer to this as a *reduced instance*.) Let β_j^* denote the value of the dual variable for constraint j in this optimal solution.
3. For each subsequent agent i, if there is an option o with non-negative gain, select the option[3] o of maximum gain, and set $z_i = gain(o)$.

We will refer to a variant of this algorithm in Section 3 as the DualBase algorithm. The intuition behind this algorithm is simple; the dual variables β_j^* can be thought of as specifying a value/size ratio necessary for an option to be selected. An optimal choice for each β_j gives an optimal solution to the packing problem; this fact is proven implicitly in the next section, where we further show that with high probability, the optimal choice β_j^* on the sample S leads to a near-optimal solution on the entire instance.[4] In the following, let $w_{\max} = \max_{i,o}\{w_{io}\}$, and let $a_{\max} = \max_{i,o,j}\{a_{ioj}\}$.

[3] Assume for simplicity that there are no ties, and so there is a unique such option. This can be effectively achieved by adding a random perturbation to the weights; we omit details from this extended abstract.

[4] Technically, the solution returned may violate some constraints by a small factor, but it easy to modify the algorithm to avoid this; see the discussion after Lemma 2.

2.1 Proof Sketch

We now sketch a proof of Theorem 2, showing that the above training-based algorithm is a $(1 - O(\varepsilon))$-approximation. (Proofs of some claims are omitted.) Let $I^* \subseteq I$ denote the set of agents i with some option o having non-negative gain, and let $\mathcal{O}^*$ denote the set of pairs $\{(i, o) \mid i \in I^*, o = \arg\max_{o \in O(i)} gain(o)\}$. We abuse notation by writing $i \in \mathcal{O}^*$ if there exists $o \in O(i)$ such that $(i, o) \in \mathcal{O}^*$. We use $\mathcal{O}^*(S)$ to denote $\mathcal{O}^* \cap S$; note that $\mathcal{O}^* - \mathcal{O}^*(S)$ represents the options selected by the algorithm (for the purposes of analysis, we do not select any options for agents in S).

Given any vector β^*, we obtain a feasible solution to **Dual-LP** by selecting for each item in I^*, the option o such that $(i, o) \in \mathcal{O}^*$ and setting $z_i = gain(o)$; for each item $i \notin I^*$, we set $z_i = 0$.

Definition 1. *Let* $W = \sum_{(i,o)\in\mathcal{O}^*} w_{io}$ *be the total weight of selected options, and let* $W(S) = \sum_{(i,o)\in\mathcal{O}^*(S)} w_{io}$. *Let* $C(j) = \sum_{(i,o)\in\mathcal{O}^*} a_{ioj}$ *and* $C(j, S) = \sum_{(i,o)\in\mathcal{O}^*(S)} a_{ioj}$.

For any *fixed* vector β^*, $\mathcal{O}^*$ and hence W and each $C(j)$ are independent of the choice of the sample S; the expected value of $W(S)$ is εW, and that of $C(j, S)$ is $\varepsilon C(j)$.[5] For fixed β^*, we can conclude that the the values of $W(S)$ and $C(j, S)$ will be close to their expectations with high probability. However, the β^* computed by the algorithm is itself a function of S; the main idea of the proof (similar to that of [10] on the special case of the AW problem) is that one can still show that if β^* satisfies the complementary slackness conditions on the first εn agents (being an optimal solution), w.h.p. it *approximately* satisfies these conditions on the entire set. Two main differences from the proof of [10] are that the structure of our LP is unrestricted, and that the weight and demand coefficients of a given option may be completely unrelated; dealing with these requires additional care, and a weaker dependence of ε on the input parameters.

Definition 2. *For a sample* S *and* $j \in [m]$, *let* $r_j(S) = |C(j, S) - \varepsilon C(j)|$, *and let* $t(S) = |W(S) - \varepsilon W|$. *When the context is clear, we will abbreviate* $r_j(S)$ *by* r_j *and* $t(S)$ *by* t.

1. *The sample* S *is* r_j-bad *if:*
   $r_j \geq (m + 1)(\ln n + \ln q)a_{\max} + \sqrt{C(j)} \cdot \left(2\sqrt{\varepsilon(m + 1)(\ln n + \ln q)a_{\max}}\right)$.
2. *The sample* S *is* t-bad *if:*
   $t \geq (m + 1)(\ln n + \ln q)w_{\max} + \sqrt{W} \cdot \left(2\sqrt{\varepsilon(m + 1)(\ln n + \ln q)w_{\max}}\right)$.

Lemma 1. $\boldsymbol{Pr}[S$ *is* r_j*-bad*$] \leq \frac{1}{m\cdot(nq)^{m+1}}$ *for each* j, *and* $\boldsymbol{Pr}[S$ *is* t*-bad*$] \leq \frac{1}{(nq)^{m+1}}$.

We argue below that if S is *not* t-bad or r_j-bad for any j, we obtain a good solution. We use the following simple proposition:

---

[5] Though β^* depends on S, many distinct samples S may lead to the same vector β^*. Also, we take expectations over *all* choices of S, not just those leading to the given β^*. We will later use the union bound over all possible vectors β^*.

Proposition 1. *Let $j \in [m]$ be a constraint such that $C(j, S) = \varepsilon$. If S is not r_j-bad, we have $1 - 2\varepsilon \leq C(j) \leq 1 + 3(\varepsilon + \varepsilon^2)$.*

Lemma 2. *If the sample S is not t-bad or r_j-bad for any constraint j, the value of the options selected by the algorithm is $(1 - O(\varepsilon))\text{OPT}$.*

Note that the options selected by the algorithm, as described above, may not be feasible even if S is not r_j-bad; Proposition 1 only implies that $C(j) \leq 1 + 3(\varepsilon + \varepsilon^2)$. Thus, we might violate some constraints by a small amount. This is easily fixed: simply decrease the capacities of all constraints by a factor of $1 + O(\varepsilon)$. This reduces the value of the optimal solution by no more than the same factor; though our algorithm might violate the reduced capacity of constraint j by a factor of $1 + O(\varepsilon)$, we respect the original capacity when S is not r_j-bad. Thus, when S is not t-bad or r_j-bad for any j, we obtain a feasible solution with value $(1 - O(\varepsilon))\text{OPT}$.

Finally, Lemma 1 implies that for any fixed β^*, the probability that a random sample S of agents is bad is less than $\frac{2}{(nq)^{m+1}}$. The following lemma shows that there are at most $(nq)^m$ distinct choices for β^*; as a result, the sample is good for any β^* with high probability. Therefore, with high probability, our algorithm returns a feasible solution with value at least $(1-O(\varepsilon))\text{OPT}$, proving Theorem 2.

Lemma 3. *There are fewer than $(nq)^m$ distinct solutions β^* that are returned by the algorithm after step 2.*

3 Experimental Evaluation: Efficiency and Fairness

We now present experimental results in the Display Ad setting, evaluating our algorithm and comparing its performance, in terms of efficiency and *fairness*,to other online algorithms and also to "ideal" solutions computed offline.

Our data set consists of a uniform sample of a set of impressions and a set of advertisers for six different publishers (A-F) from one week in September 2009. The number of arriving impressions varies from 200k to 1,500k impressions. The number of advertisers per publisher varies from 100 to 2,600 advertisers. Each impression is tagged with its set of eligible advertisers and an *edge weight* for each eligible advertiser capturing the "targeting quality" for this advertiser. We compare the algorithms both in terms of the social efficiency and the *fairness* of their solutions.

Fairness in Ad Allocation. Besides efficiency, fairness plays an important role in measuring the performance of an ad allocation. An allocation that achieves large total value while delivering very few impressions to some advertisers is undesirable; advertisers whose contracts are unfulfilled must typically be paid a penalty, and the publisher may have trouble retaining such advertisers.

Different notions of fairness in resource allocation have been explored extensively in the literature [20,22,16,23]; perhaps the most common of these is *max-min* fairness. Such a metric is not appropriate in the DA setting, where the contract of one advertiser may specify many fewer impressions than that of others, and thus the total value he obtains *should* be lower.

For a detailed discussion of fairness metrics, see the full version of our paper at http://arxiv.org/abs/1001.5076 [11]. Briefly, given an allocation x, we define its *value vector* $v(x)$ such that the jth component is the total value obtained by advertiser j. We roughly define the fairness metric as the l_1 distance between the value vector of x and that of an (appropriately normalized) ideal allocation x^*. In the full version [11], we consider various candidate ideal allocations; the one we focus on (fractionally) allocates an impression by dividing it equally among all advertisers who are *interested* in it; advertisers are interested in as many of the impressions they value highly as needed to satisfy their contracts.

The Algorithms. We examine (a) 3 pure online algorithms, (b) 2 training-based online algorithms, and (c) 2 offline algorithms.

(a) The pure online algorithms are GREEDY, PD_AVG, and PD_EXP, that are developed and analyzed in a previous paper [12], and achieve worst-case competitive ratios of $\frac{1}{2}$, $\frac{1}{2}$, and $1 - \frac{1}{e}$. These algorithms are primal-dual algorithms that proceed as follows: we compute a discounting factor β_a based on the set of impressions already assigned to advertiser a, and then upon arrival of a new impression i, we assign this impression i to an advertiser a maximizing $w_{ia} - \beta_a$. The difference between these algorithms is in computing β_a: Let $w_1, w_2, \ldots w_{n(a)}$ be the weights of impressions currently assigned to advertiser a, sorted in non-increasing order. In GREEDY, PD_AVG, and PD_EXP, we set $\beta_a = w_{1a}$, $\beta_a = \frac{\sum_{1\le j\le n(a)} w_j}{n(a)}$, and $\beta_a = \frac{1}{n(a)\cdot\left((1+1/n(a))^{n(a)}-1\right)} \sum_{j=1}^{n(a)} w_j \left(1 + \frac{1}{n(a)}\right)^{j-1}$, respectively.

(b) The training-based online algorithms are the dual-based algorithm DualBase from Section 2 and a HYBRID algorithm in which we set β_a for each advertiser a to be a convex combination of the DualBase and PD_AVG algorithms, i.e., we start using β_a from DualBase, and then slowly move to using β_a from PD_AVG. (The hybrid algorithm is inspired by ideas from [24].) To train and test them we used for each data set a random sample of 1% of the impressions for training and the remaining 99% for testing. This proxies the random order model, where a sample from the beginning part of the sequence is equivalent to a sample of the whole data.

(c) The offline algorithms are the fair algorithm FAIR using *equal sharing* described above, and the optimum efficient offline algorithm LP_WEIGHT.

Experimental Results. The normalized efficiency and normalized fairness of each of the algorithms are summarized in Table 1. Regarding efficiency it shows that (1) the training-based algorithms perform very similarly (except for one publisher) and outperform the pure online algorithms (5-12% improvement), (2) of the pure online algorithms, both PD_AVG and PD_EXP outperform GREEDY, even though both PD_AVG and GREEDY are 1/2-competitive in the worst case, (3) PD_EXP shows only a 5% overall improvement over PD_AVG, even though the worst-case competitive analysis of PD_EXP is much better than PD_AVG. Since the value of fairness depends on the values assigned to advertisers and different publishers have different advertisers, we normalized the fairness values for each publisher so that the *least* fair algorithm achieves a score of 100 and the best achieves a score of 0. Normalizing allows us to compute the average over

Table 1. Normalized efficiency and fairness of different algorithms for different publishers and averaged over all publishers. All numbers are normalized between 0 and 100 such that the efficiency of OPT =LP_WEIGHT is 100 and 0 is the most fair solution.

	Normalized Efficiency							Normalized Fairness						
Publishers	A	B	C	D	E	F	Avg	A	B	C	D	E	F	Avg
LP_WEIGHT	100	100	100	100	100	100	100	34.6	47.7	98.8	100	70.3	90.1	73.6
FAIR	88.2	98.4	73.6	42.3	74.6	53.3	71.7	0	0	0	0	0	0	0
DualBase	85	93	85.7	74	91.8	93.5	87.2	69.5	62.5	96.7	43.1	87.9	88.6	74.7
HYBRID	85	93.8	95.2	73.8	92.7	93.5	89	69.4	63.1	100	41.9	83.7	88.6	74.5
PD_AVG	72	93.2	75.3	65.3	71.7	89.5	77.8	73	72	82.7	31.7	91.9	85.3	72.8
PD_EXP	72.6	89.7	73.9	90.8	72.6	96.3	82.6	69.7	59.5	86.1	71	88.8	100	79.2
GREEDY	64	90.5	69.7	53.6	55	86.2	69.8	100	100	98.6	45	100	100	90.6

different publishers. The results in the table indicate that GREEDY is the least fair algorithm and the remaining algorithms perform roughly the same, though their performance differs over different publishers. For a more detailed analysis of the results see the appendix.

4 Concluding Remarks

This paper motivates many open problems to explore: (i) Can we achieve an algorithm that is simultaneously good both in the worst case and in stochastic settings? Such an algorithm would be of use when the actual distribution of agents is different from the one predicted/learnt from a sample; in the display ad setting, this occurs when there is a sudden spike in traffic to a website, perhaps in response to a breaking news event, or links from an extremely high-traffic source. (ii) Can we design an online allocation algorithm that provably achieves approximate efficiency and approximate fairness (for some an appropriate notion of fairness) at the same time? (iii) Can we prove that in certain settings that appear in practice, the PD_AVG algorithm achieves an improved approximation factor (i.e., better than $\frac{1}{2}$)? (iv) Can we extend the online stochastic algorithm studied in this paper to other stochastic process models such as Markov-based stochastic models? Answering these questions is an interesting subject of future research.

Acknowledgments. This paper is a followup of our previous work with S. Muthukrishnan and Martin Pál, and some of the results and discussions in this paper are inspired by our initial discussions with them. We thank Martin and Muthu for their contributing insights toward this paper. We also thank the Google display ad team, and especially Scott Benson for helping us with data sets used in this paper.

References

1. Agrawal, S., Wang, Z., Ye, Y.: A dynamic near-optimal algorithm for online linear programming, Working paper posted at `http://www.stanford.edu/~yyye/`
2. Alaei, S., Malekian, A.: Maximizing sequence-submodular functions (2009) (manuscript)

3. Awerbuch, B., Azar, Y., Plotkin, S.: Throughput-competitive on-line routing. In: FOCS, vol. 34, pp. 32–40 (1993)
4. Azar, Y., Birnbaum, B., Karlin, A., Mathieu, C., Nguyen, C.: Improved Approximation Algorithms for Budgeted Allocations. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 186–197. Springer, Heidelberg (2008)
5. Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R.: Online auctions and generalized secretary problems. SIGecom Exchanges 7(2) (2008)
6. Buchbinder, N., Jain, K., Naor, J.: Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, p. 253. Springer, Heidelberg (2007)
7. Buchbinder, N., Naor, J.: Improved bounds for online routing and packing via a primal-dual approach. In: FOCS, pp. 293–304 (2006)
8. 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)
9. Chawla, S., Hartline, J.D., Malec, D., Sivan, B.: Sequential posted pricing and multi-parameter mechanism design. In: Proc. of ACM STOC, pp. 311–320 (2010)
10. Devanur, N., Hayes, T.: The adwords problem: Online keyword matching with budgeted bidders under random permutations. In: ACM EC (2009)
11. Feldman, J., Henzinger, M., Korula, N., Mirrokni, V.S., Stein, C.: Online stochastic ad allocation: Efficiency and fairness, http://arxiv.org/abs/1001.5076
12. Feldman, J., Korula, N., Mirrokni, V., Muthukrishnan, S., Pal, M.: Online ad assignment with free disposal. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 374–385. Springer, Heidelberg (2009)
13. Feldman, J., Mehta, A., Mirrokni, V., Muthukrishnan, S.: Online stochastic matching: Beating 1 - 1/e. In: FOCS (2009)
14. Ghosh, A., McAfee, P., Papineni, K., Vassilvitskii, S.: Bidding for representative allocations for display advertising. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 208–219. Springer, Heidelberg (2009)
15. Ghosh, A., Rubinstein, B.I.P., Vassilvitskii, S., Zinkevich, M.: Adaptive bidding for display advertising. In: WWW, pp. 251–260 (2009)
16. Goel, A., Meyerson, A., Plotkin, S.A.: Combining fairness with throughput: online routing with multiple objectives. In: STOC, pp. 670–679 (2000)
17. Goel, G., Mehta, A.: Adwords auctions with decreasing valuation bids. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 335–340. Springer, Heidelberg (2007)
18. Goel, G., Mehta, A.: Online budgeted matching in random input models with applications to adwords. In: SODA, pp. 982–991 (2008)
19. Karp, R., Vazirani, U., Vazirani, V.: An optimal algorithm for online bipartite matching. In: Proc. STOC (1990)
20. Kleinberg, J.M., Rabani, Y., Tardos, É.: Fairness in routing and load balancing. J. Comput. Syst. Sci. 63(1), 2–20 (2001)
21. Korula, N., Pal, M.: Algorithms for secretary problems on graphs and hypergraphs. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5556, pp. 508–520. Springer, Heidelberg (2009)
22. Kumar, A., Kleinberg, J.M.: Fairness measures for resource allocation. SIAM J. Comput. 36(3), 657–680 (2006)
23. Lipton, R., Markakis, E., Mossel, E., Saberi, A.: On approximately fair allocations of indivisible goods. In: ACM EC (2004)

24. Mahdian, M., Nazerzadeh, H., Saberi, A.: Allocating online advertisement space with unreliable estimates. In: ACM EC, pp. 288–294 (2007)
25. Mehta, A., Saberi, A., Vazirani, U., Vazirani, V.: Adwords and generalized online matching. In: FOCS (2005)
26. 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)
27. Vee, E., Vassilvitskii, S., Shanmugasundaram, J.: Optimal online assignment with forecasts. In: Proc. of ACM EC, pp. 109–118 (2010)