Online Dual Decomposition for Performance and Delivery-Based Distributed Ad Allocation

Jim C. Huang
Amazon
Seattle, WA, USA
huangjim@amazon.com

Rodolphe Jenatton
Amazon
Berlin, Germany
jenatton@amazon.com

Cédric Archambeau
Amazon
Berlin, Germany
cedrica@amazon.com

ABSTRACT

Online optimization is central to display advertising, where we must sequentially allocate ad impressions to maximize the total welfare among advertisers, while respecting various advertiser-specified long-term constraints (e.g., total amount of the ad's budget that is consumed at the end of the campaign). In this paper, we present the online dual decomposition (ODD) framework for large-scale, online, distributed ad allocation, which combines dual decomposition and online convex optimization. ODD allows us to account for the distributed and the online nature of the ad allocation problem and is extensible to a variety of ad allocation problems arising in real-world display advertising systems. Moreover, ODD does not require assumptions about auction dynamics, stochastic or adversarial feedback, or any other characteristics of the ad marketplace. We further provide guarantees for the online solution as measured by bounds on cumulative regret. The regret analysis accounts for the impact of having to estimate constraints in an online setting before they are observed and for the dependence on the smoothness with which constraints and constraint violations are generated. We provide an extensive set of results from a large-scale production advertising system at Amazon to validate the framework and compare its behavior to various ad allocation algorithms.

Keywords

distributed optimization; display advertising; campaign optimization; ad allocation; budget pacing; long-term constraints; demand-side platform

1. INTRODUCTION

The past few years have seen significant interest and research in ad allocation in online display advertising, where the problem consists of serving ad impressions to users from many competing ads across a large number of websites and mobile apps, subject to a variety of advertiser objectives and constraints. For example, advertisers typically expect that an advertising platform will maximize ad performance (i.e., the ad's effectiveness at driving user behavior), subject to constraints on user targeting and constraints on ad delivery, such that the advertising platform is expected to spend as close to 100% of an ad's budget as possible by the end of the ad's flight time under a cost-per-mille, or CPM, billing model. In the literature on display advertising [3, 6, 17], online ad allocation is typically formulated as an online optimal assignment problem over a bi-partite graph in which nodes correspond to both ads and bid requests (i.e., user visits to a web page). Bid requests are processed sequentially, with edges corresponding to possible assignments of an ad to a bid request with weight equal to the advertiser's welfare for the given ad/bid request pair. In practice, the ad allocation problem is characterized by being 1) online, and 2) distributed (i.e., variable updates must be executed in parallel across large fleets of machines), with 3) potentially multiple long-term constraints that require estimation of quantities at bid request time (e.g., estimating the cost of an ad impression before winning it on external exchanges).

Various algorithms have been proposed for the ad allocation problem in the online display advertising literature [3, 6, 9, 10], but these have largely been derived as either approximation algorithms or heuristics, increasing the difficulty and complexity in extending these methods to a variety of constraints [17]. The distributed aspect of online display advertising has not been explicitly addressed either, especially in the presence of long-term constraints, which leads to further approximations in a production setting. Moreover, algorithms proposed have not accounted for highly variable, dynamic constraint violations (e.g., spend amounts) per ad that are often encountered in practical online ad serving systems. This is one reason why ad allocation solved as an offline problem can often be sub-optimal in practice [6]. Finally, the impact of having to estimate constraints in order to allocate ad impressions (e.g., in real-time bidding, or RTB, exchanges, the clearing price of an ad auction is only revealed to us after we've decided on which ad to show).

To the best of our knowledge, a comprehensive and holistic framework that satisfactorily addresses the above challenges faced in practical web-scale algorithms is still missing. In this paper, we present a general framework for deriving online and distributed ad allocation algorithms for dealing with a variety of online ad allocation problems in display advertising. Our contributions can be summarized as follows:

- We introduce a framework to derive online distributed optimization algorithms applicable to online ad alloca-

KDD '16, August 13 - 17, 2016, San Francisco, CA, USA

ISBN 978-1-4503-4232-2/16/08...$15.00

DOI: http://dx.doi.org/10.1145/2939672.2939691

tion problems with long-term advertiser-specified constraints;

- We account for the impact of having to estimate constraints before they are observed, without explicitly assuming bandit feedback;
- Leveraging the theoretical analysis provided in [12], we show that our approach enjoys bounds on dynamic cumulative regret, which depend on the smoothness with which constraint violations and the constraints themselves occur to the oracle learner over time.These results do not make assumptions about auction dynamics, stochastic or adversarial feedback, or other characteristics of the ad marketplace.
- We validate the framework by reporting results on traffic data collected by a large-scale display advertisement system at Amazon.

We call our framework online dual decomposition (ODD), which allows us to efficiently solve general large-scale optimization problems in a distributed and online fashion.

2. ONLINE DISPLAY ADVERTISING AS AN OPTIMAL ASSIGNMENT PROBLEM

In online advertising, each user visit to a web page triggers a bid request i to be sent in real-time (through possibly complex channels) to a host, or machine, in an ad serving fleet. Each incoming bid request has a set of candidate ads that can be served: the set of candidates that are eligible to be shown for a bid request are determined by several advertiser-specified constraints such as behavioral or demographic targeting (i.e., an ad is to be/not to be shown only to users satisfying targeting criteria), or frequency capping (i.e., an ad is not to be shown more than some number of times to a given unique user over some time period), to name a few. Indexing bid requests by i, we denote by $C_i \subseteq \{1, \cdots, M\}$ the subset of M ads that are eligible to be shown for bid request i as a result of such constraints. Indexing ads by $j = 1, \cdots, M$, the welfare of serving an impression for ad j (i.e., the value of showing the ad once to a user) for bid request i is given by $v_{ij} \geq 0$. The welfare v_{ij} can be set to the expected value of serving one ad impression for ad j for bid request i, but other expressions for welfare may be used as well.

For bid request i, let assignment variable $x_{ij} = 1$ correspond to our decision to allocate ad j an impression, with $x_{ij} = 0$ otherwise. Let w_i be the clearing price associated with allocating an impression for bid request i (e.g., the result of a second-price auction on external ad exchanges). Similarly, let a_{ij} be the some advertising quantity (e.g., budget consumed) incurred a result of allocating an impression for bid request i for ad j, where we generally assume that advertisers are charged per impression (as opposed to, say, a cost-per-click or cost-per-action model). We note at this juncture that a_{ij} and the clearing price w_i can be distinct in practice. In particular, the method by which a_{ij} is computed can vary from ad to ad and from one bid request to the next, depending on contracts that advertisers may hold with the ad serving system.

Given the welfare v_{ij} of serving an impression of ad j for bid request i, we wish to derive an online and distributed algorithm for selecting ads j such that in aggregate, we maximize total welfare across all advertisers subject to long-term constraints specified by the advertisers as a function of all variables x_{ij}. We will also have the constraints that for any bid request i, $\sum_{j \in C_i} x_{ij} \leq 1$, such that we can show at most one ad per bid request (we can also elect to show none). As we assume a distributed and online setting, bid requests are divided across time and among several hosts. Thus, there are two ways to partition bid requests: the first is by time, whereby we can divide the continuous time axis into rounds of T approximately even-sized intervals indexed by $t = 1, \cdots, T$. For a given interval t, let I_t denote the set of bid requests processed across the entire fleet in that interval, such that the sets I_t form a partition of all bid requests. Given the above, let $\mathbf{x}_t, \mathbf{u}_t$ denote assignment and utility[1] vectors whose elements correspond to $x_{ij}, v_{ij} - w_i$ respectively, and define matrices $\mathbf{A}_t$ similarly. For illustration, Figure 1 shows a toy example of an assignment problem with scalar quantities and corresponding matrices and vectors as defined above.

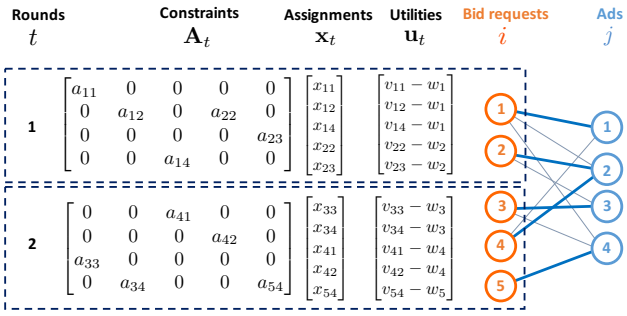


Figure 1: A toy assignment problem consisting of 4 ads and 5 bid requests divided into 2 rounds, with corresponding matrix and vector quantities $\mathbf{A}_t, \mathbf{x}_t, \mathbf{u}_t$. Here, elements a_{ij} correspond to some advertising output of interest (e.g., budget consumed) for bid request i and ad j, with $x_{ij} \in \{0, 1\}$. Dark edges in the bipartite graph on the right correspond to a possible feasible assignment. Figure best seen in color.

As has been commonly done in online display advertising [3, 6, 9], we formulate the canonical problem of allocating ad impressions as an optimal assignment problem, where the goal is to match ads to bid requests such that we maximize the total difference between welfare and clearing price (profit) across all ads and all bid requests for the ad serving platform. The standard optimal assignment problem can be written as a linear program (LP) relaxation of the corresponding combinatorial optimization problem with assignment variables $\mathbf{X} = \{x_{ij}\}$:

$$\underset{\mathbf{X}}{\text{maximize}} \sum_{i,j \in C_i} (v_{ij} - w_i) x_{ij} \quad \text{with} \quad \begin{cases} \forall\, i, \sum_{j \in C_i} x_{ij} \leq 1 \\ \forall\, i, j \in C_i,\ x_{ij} \geq 0. \end{cases}$$

We now introduce the online aspect of the problem (we postpone discussion of the distributed aspects for later) by partitioning bid requests into sets I_t for $t \in \{1, \cdots, T\}$.

[1]Utility measures the advertiser's preference between bid requests.

Using matrix-vector notation and introducing the quantity $N \equiv \max_t M|I_t|$, we can compactly re-write the optimal assignment problem as $\underset{\mathbf{x}_1 \in \mathcal{X}_1, \cdots, \mathbf{x}_T \in \mathcal{X}_T}{\text{maximize}} \sum_{t=1}^{T} \mathbf{u}_t^\top \mathbf{x}_t$, where we have defined

$$\mathcal{X}_t \equiv \Big\{ \mathbf{x} \in [0,1]^N \Big| \forall\, i \in I_t, \sum_{j \in C_i} x_{ij} \leq 1 \text{ and } x_{ij} = 0 \text{ if } j \notin C_i; \\ \forall i \notin I_t, x_{ij} = 0 \Big\},$$

and $\mathbf{x}_t$ is a row in the matrix $\mathbf{X} \in \{0,1\}^{T \times N}$. In the online assignment problem, for round t the external world generates the vector $\mathbf{u}_t$, after which we must make allocation decisions $\mathbf{x}_t$ and obtain reward $\mathbf{u}_t^\top \mathbf{x}_t$. Thus, each element of vector $\mathbf{x}_t \in \mathcal{X}_t$ is an assignment variable x_{ij}, with the elements of vector $\mathbf{u}_t$ similarly ordered to be equal to $v_{ij} - w_i$.

It has been shown [6] that we can solve the optimal assignment problem iteratively for each bid request i by sorting ads $j \in C_i$ by v_{ij}, picking the top-ranking ad and assigning an impression to that ad [6] if $\max_j v_{ij} \geq w_i$. For completeness and due to its central role in what follows, we formalize this result as a Lemma and reproduce the proof (which can also be found in [6]) in the Appendix.

LEMMA 1. *Denote by $\mathbf{x}_t$ to be the set of all x_{ij}'s that are processed in round t. Let $j^\star = \text{argmax}_{j \in C_i} v_{ij}$, and let β_i be a dual variable for enforcing the constraint $\sum_{j \in C_i} x_{ij} \leq 1$. Then for each $i \in I_t$, we must have*

$$\beta_i^\star = v_{ij^\star} = \max_{j \in C_i} v_{ij},$$
$$\hat{x}_{ij^\star} = 1, \hat{x}_{ij} = 0 \; \forall j \in C_i, j \neq j^\star \text{ if } \beta_i^\star \geq w_i,$$
$$\hat{x}_{ij} = 0 \; \forall j \text{ if } \beta_i^\star < w_i,$$

such that $\hat{\mathbf{x}}_t = \text{argmax}_{\mathbf{x}_t \in \mathcal{X}_t} \mathbf{u}_t^\top \mathbf{x}_t$.

Thus, for a given round t, the above online algorithm sequentially allocates impressions such that the optimum $\hat{\mathbf{x}}_t \in \text{argmax}_{\mathbf{x}_t \in \mathcal{X}_t} \mathbf{u}_t^\top \mathbf{x}_t$ is attained. In particular, the quantity $\beta_i^\star$ plays the role of the bid submitted to an auction for a given ad impression opportunity within a larger advertising marketplace. Consistent with the practical reality of ad allocation, if $\beta_i^\star < w_i$, we do not allocate any ad impression (i.e., we lost the auction) and otherwise we get to allocate an ad impression to the winning ad $j^\star \in C_i$. Thus, the calculation of $\hat{\mathbf{x}}_t$ is wholly determined by computing bids $\beta_i^\star$ and finding whether we win an ad impression or not via the clearing price w_i for bid requests $i \in I_t$.

In the sequel, we take advantage of the above result to introduce *concave* functions of $\mathbf{X}$ for modelling long-term constraints and solve the resulting non-linear maximization problems via an online primal-dual method (with guarantees on the resulting solution as compared to the optimal offline solution), whilst preserving the computationally-efficient primal steps with some simple modifications. Under our proposed framework, we will modify each bid as a function of long-term constraints to $\beta_i^\star = \max_j \{v_{ij} - \lambda_{j,t} a_{ij}\}$ such that $\hat{\mathbf{x}}_t = \text{argmax}_{\mathbf{x}_t \in \mathcal{X}_t} \{\mathbf{u}_t^\top \mathbf{x}_t - \boldsymbol{\lambda}^\top \mathbf{A}_t \mathbf{x}_t\}$ for some dual variables $\boldsymbol{\lambda}$ (formally defined in the sequel) that allow us to penalize violations of constraints. Our framework has the property that no knowledge of auction dynamics, assumptions about stochastic or adversarial feedback, or knowledge of other characteristics of the ad marketplace are needed to provide regret guarantees for online ad allocation. Moreover, as we will show, the resulting algorithm can be naturally extended to a distributed setting in which we must allocate ad impressions across a large number of ad serving hosts operating independently, which is key for practical large-scale ad serving systems.

3. OPTIMAL ASSIGNMENT WITH LONG-TERM CONSTRAINTS

As we will show, many online optimization problems of interest in online display advertising will consist of an optimal assignment problem with additional equality and inequality constraints on the primal variables $\mathbf{x}_t$, which introduces non-trivial challenges. We assume that these constraints are 1) measured over T rounds, and 2) are linear in the assignment variables $\mathbf{X}$, such that we can account for such constraints by penalizing some measure of $\frac{1}{T}\sum_{t=1}^{T} \mathbf{A}_t \mathbf{x}_t - \mathbf{b}_t$, where $\mathbf{A}_t \in \mathcal{A} \subseteq \mathbb{R}^{K \times N}, \mathbf{b}_t \in \mathbb{R}^K$. Building on the work from [12], the optimal assignment problem can be modified to:

$$\underset{\mathbf{x}_1 \in \mathcal{X}_1, \cdots, \mathbf{x}_T \in \mathcal{X}_T}{\text{maximize}} \; \frac{1}{T} \sum_{t=1}^{T} \mathbf{u}_t^\top \mathbf{x}_t - \mathcal{E}\left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{A}_t \mathbf{x}_t - \mathbf{b}_t \right) \quad (1)$$

where $\mathcal{E}$ is a convex function measuring the residuals of the constraints. In this paper, we will assume that $\mathcal{E}$ is smooth in the sense that it has Lipschitz continuous gradients over the domain $\mathbb{R}^K$. We note that although the above optimization problem is not linear in all variables in the strict sense of the optimal assignment problem, the corresponding Lagrangian (and primal-dual algorithm that we will derive) remains linear with respect to assignment variables $\mathbf{X}$, which implies that the computationally efficient ranking scheme derived for the optimal assignment problem can be modified to solve for the primal variables in the above problem. Moreover, we note that unlike [2], we are not interested in the (unweighted) average of $\mathbf{x}_t$ vectors insofar as objectives and constraints are concerned, and we are explicitly interested in maximizing functions of $\mathbf{x}_t$ that are time-varying and highly dynamic. Before we continue, we illustrate two common problem classes that arise frequently in online display advertising that can be naturally formulated as optimization problems of the above form.

EXAMPLE 1. *(Max-performance, target delivery ad allocation) In this problem formulation, we wish to maximize ad performance (as measured by welfare v_{ij} for each bid request and eligible ad candidate) subject to the need to consume as close to 100% of each ad's total budget for T rounds. Denoting 1) a_{ij} to be the revenue charged to the advertiser for ad j upon serving an impression for ad j and bid request i and 2) b_j to be ad j's total budget to be delivered over T rounds, we have $K = M$ (one constraint per ad for M ads), $\mathbf{A}_t \in \mathbb{R}^{M \times N}$ is a matrix whose $(j,i)^{th}$ element is a_{ij} and $\mathbf{b}_t \in \mathbb{R}^M$ is a vector whose j^{th} element is b_j, so that $\frac{1}{T}\sum_{t=1}^{T} \mathbf{A}_t \mathbf{x}_t - \mathbf{b}_t$ is the average error between the amount of budget consumed and the target amount of budget to be consumed over T rounds, which we wish to penalize via a suitable choice of $\mathcal{E}$.*

EXAMPLE 2. *(Max-delivery, max cost-per-action ad allocation) Here, we aim to maximize welfare subject to a constraint on the cost-per-action (CPA) for each ad, specified as $c_j > 0$. With $K = M$, let p_{ij} be the probability of user conversion, and let r_{ij} be the amount of revenue charged to*

the advertiser upon serving an ad impression for bid request i and ad j. Given the quantities defined in Example 1, we can formulate the CPA constraint for ad j as

$$\forall\, j, \sum_i r_{ij} x_{ij} \leq c_j \sum_i p_{ij} x_{ij}.$$

This can be formulated equivalently as $\frac{1}{T}\sum_{t=1}^{T} \mathbf{A}_t \mathbf{x}_t - \mathbf{b}_t$ where $\mathbf{A}_t$ is a matrix whose $(j,i)^{th}$ element is $(r_{ij} - c_j p_{ij})$ and $\mathbf{b}_t = \mathbf{0}$ so that penalizing values of $\frac{1}{T}\sum_{t=1}^{T} \mathbf{A}_t \mathbf{x}_t - \mathbf{b}_t$ above $\mathbf{0}$ is equivalent to a penalty for violating the CPA constraint for each ad.

A list of possible instantiations for $\mathcal{E}$ for the above problems can be found in [12]. We highlight that in the general case, we allow $K \neq M$, which allows us to account for multiple constraints per ad, and/or diverse sets of constraints for different ads.

Consider now the Lagrangian, or saddle-point function, for the problem in (1), given by

$$\mathcal{L}(\mathbf{X}, \boldsymbol{\lambda}) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{u}_t^\top \mathbf{x}_t - \boldsymbol{\lambda}^\top \left[\frac{1}{T}\sum_{t=1}^{T} \mathbf{A}_t \mathbf{x}_t - \mathbf{b}_t\right] + \mathcal{E}^\star(\boldsymbol{\lambda}), \quad (2)$$

where $\boldsymbol{\lambda} \in \Lambda$ are dual variables that belong to the domain $\Lambda \subseteq \mathbb{R}^K$ of $\mathcal{E}^*$, where $\mathcal{E}^\star(\boldsymbol{\lambda}) \equiv \sup_{\mathbf{z}\in\mathbb{R}^K}\{\boldsymbol{\lambda}^\top \mathbf{z} - \mathcal{E}(\mathbf{z})\}$ is the Fenchel conjugate [4] of $\mathcal{E}$ for which we assume the domain is compact so that there exists $R_\lambda \equiv \max_{\boldsymbol{\lambda}\in\Lambda} \|\boldsymbol{\lambda}\|_2 < +\infty$. Moreover, we remark that $\mathcal{E}^\star$ is strongly convex by virtue of $\mathcal{E}$ having Lipschitz continuous gradients [14]. We recall that a function $\mathcal{E}^\star$ is strongly convex with modulus $\sigma > 0$ if $\mathcal{E}^\star(\mathbf{u}) \leq \mathcal{E}^\star(\mathbf{v}) + \nabla\mathcal{E}^\star(\mathbf{u})^\top(\mathbf{u}-\mathbf{v}) - \frac{\sigma}{2}\|\mathbf{u}-\mathbf{v}\|^2$ for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^K$. We can rewrite the Lagrangian function described in (2) as $\mathcal{L}(\mathbf{X}, \boldsymbol{\lambda}) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})$, where

$$\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) = \begin{cases} \mathbf{u}_t^\top \mathbf{x}_t - \boldsymbol{\lambda}^\top\Big(\mathbf{A}_t\mathbf{x}_t - \mathbf{b}_t\Big) + \mathcal{E}^\star(\boldsymbol{\lambda}) & \text{if } \boldsymbol{\lambda} \in \Lambda \\ -\infty & \text{otherwise.} \end{cases}$$

To maximize $\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ with respect to $\mathbf{x}_t$, it is straightforward to show that the result of Lemma 1 can be modified to accomplish this by ranking ads for each bid request, but using a modified bid of $v_{ij} - \lambda_{j,t} a_{ij}$. Thus, in the sequel we will appeal to Lemma 1 for a computationally efficient method for solving the problem of computing $\hat{\mathbf{x}}_t \in \text{argmax}_{\mathbf{x}_t\in\mathcal{X}_t}\, \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})$.

In an offline setting where we would have the ability to iteratively update both primal variables $\mathbf{x}_t, t = 1, \cdots, T$ and dual variables $\boldsymbol{\lambda}$, the following canonical[2] primal-dual algorithm would allow us to solve for optimal variables $\mathbf{x}_t^\star$, $\boldsymbol{\lambda}^\star$ by repeating the two steps below until some convergence criterion is met:

- (P-offline) Compute for all $t = 1, \cdots, T$

$$\hat{\mathbf{x}}_t \in \arg\max_{\mathbf{x}_t\in\mathcal{X}_t} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}).$$

- (D-offline) Given $\hat{\mathbf{x}}_t$ for all $t = 1, \cdots, T$, compute $\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda})$ and gradient $\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\hat{\mathbf{X}}, \boldsymbol{\lambda}) = \sum_t \nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda})$. Update all dual variables $\boldsymbol{\lambda}$ using a projected gradient descent method with step size η. Note that we speak about a project *gradient* scheme (as opposed to a projected *subgradient* method) since the choice of $\mathcal{E}$ we will make in the experiments leads us to consider a function $\mathcal{E}^*$ which is not only strongly convex, but also differentiable on its domain Λ.

[2]There are in fact several primal-dual algorithms that could be used to solve the offline problem; we present a simple method to best illustrate the online method that follows.

The above algorithm is an offline algorithm that iteratively updates primal and dual variables until convergence. However, in practice, ad allocation requires us to make decisions in online, or sequential, fashion, such that the primal and dual steps above must be interleaved over rounds $t = 1, \cdots, T$. We now turn to formulating the ad allocation problem as an online convex optimization problem in which we sequentially update primal and dual variables $\hat{\mathbf{x}}_t$ and $\hat{\boldsymbol{\lambda}}_t$ at each round.

4. OPTIMAL AD ALLOCATION AS ONLINE CONVEX OPTIMIZATION

To move to an online convex optimization formulation, we must also account for the fact that in an online setting, not only must we update primal and dual variables sequentially, but also the constraint matrices $\mathbf{A}_t$ are only observed after we allocate ad impressions for round t (a practical example of this is that we often only find out how much to charge the advertiser once we have won an external RTB auction for a given winning ad). This requires us to estimate such matrices via $\hat{\mathbf{A}}_t$ in order to produce primal variable updates. To this end, we introduce

$$\hat{\mathcal{L}}_t(\mathbf{x}_t, \boldsymbol{\lambda}) = \mathbf{u}_t^\top \mathbf{x}_t - \boldsymbol{\lambda}^\top(\hat{\mathbf{A}}_t\mathbf{x}_t - \mathbf{b}_t) + \mathcal{E}^\star(\boldsymbol{\lambda})$$

as the Lagrangian function for round t using the estimate $\hat{\mathbf{A}}_t$ such that $\hat{\mathcal{L}}_t(\mathbf{x}_t, \boldsymbol{\lambda}) = \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}) - \boldsymbol{\lambda}^\top(\hat{\mathbf{A}}_t - \mathbf{A}_t)\mathbf{x}_t$. Then, an online version of the primal-dual method for each round $t = 1, \cdots, T$ can be described as:

- (P-online) Compute $\hat{\mathbf{x}}_t \in \text{argmax}_{\mathbf{x}_t\in\mathcal{X}_t}\, \hat{\mathcal{L}}_t(\mathbf{x}_t, \hat{\boldsymbol{\lambda}}_t)$.
- (D-online) From $\hat{\mathbf{x}}_t$, compute $\mathcal{L}_t(\hat{\mathbf{x}}_t, \hat{\boldsymbol{\lambda}}_t)$. Given gradients $\nabla\mathcal{L}_t(\hat{\mathbf{x}}_t, \hat{\boldsymbol{\lambda}}_t)$, update dual variables $\hat{\boldsymbol{\lambda}}_{t+1}$ using a projected online gradient descent method as

$$\hat{\boldsymbol{\lambda}}_{t+1} = \Pi_\Lambda\Big[\hat{\boldsymbol{\lambda}}_t - \eta_t \nabla\mathcal{L}_t(\hat{\mathbf{x}}_t, \hat{\boldsymbol{\lambda}}_t)\Big] \text{ with step size } \eta_t.$$

- (A-online) Given $\mathbf{A}_t, \hat{\mathbf{A}}_t$, update $\hat{\mathbf{A}}_{t+1}$ via a projected online subgradient method with step size ν_t.

In the above online optimization problem, we require estimating constraint matrices for each round via $\hat{\mathbf{A}}_t$ *before* we provide primal variable estimates $\hat{\mathbf{x}}_t$. This implies that $\hat{\mathbf{x}}_t$ is obtained from a perturbation of $\mathcal{L}_t(\mathbf{x}_t, \hat{\boldsymbol{\lambda}}_t)$ via $\hat{\mathcal{L}}_t(\mathbf{x}_t, \hat{\boldsymbol{\lambda}}_t)$, with $\mathbf{A}_t$ observed only *after* we have made the ad assignment $\hat{\mathbf{x}}_t$.

Estimating $\mathbf{A}_t$: To estimate constraint matrices $\mathbf{A}_t$ online, we assume that in addition to the dual variables being bounded in ℓ_2 norm with radius R_λ, we also have $\|\mathbf{x}_t\|_2 \leq R_x$ for all $t, \mathbf{x}_t \in \mathcal{X}_t$. Similarly, let R_A be such that $\|\mathbf{A}\|_F \leq R_A$ for all $\mathbf{A} \in \mathcal{A}$ and $\mathcal{A}$ is convex. We further assume that we perform an online projected subgradient method on matrices $\hat{\mathbf{A}}_t$ to estimate the sequence $\mathbf{A}_1, \ldots, \mathbf{A}_T$ where $\hat{\mathbf{A}}_{t+1} = \Pi_{\mathcal{A}}[\hat{\mathbf{A}}_t - \nu_t \mathbf{G}_t]$, with $\mathbf{G}_t$ a subgradient of $\mathbf{A} \mapsto \|\mathbf{A}_t - \mathbf{A}\|_F$ at $\hat{\mathbf{A}}_t$. With the step size $\nu_t = R_A/\sqrt{t}$, it can be shown [12] that for any sequence $\{\mathbf{A}_t\}_{t=1}^T$, the overall cumulative regret bound for the estimation of $\mathbf{x}_t$'s is additively impacted by

the term

$$\frac{6R_\lambda R_x}{\sqrt{T}}\left[\sum_{t=1}^{T}\|\mathbf{A}_t-\mathbf{A}_{t+1}\|_F+R_A\right],$$

where $\|\cdot\|_F$ stands for the Frobenius norm. This contribution captures the variations in the constraints $\mathbf{A}_t$ themselves from one round t to the next, plus a function of the largest constraint matrix (as measured by R_A) observed over the course of the algorithm. With the above result and having described our online algorithm, we are now ready to present cumulative regret bounds for the above algorithm as measured by the difference in the primal objectives for the oracle solution and the online solution obtain from the above online algorithm, where both solutions account for advertiser objectives and long-term constraints. That is, we will compare the performance of the online algorithm to that of an oracle algorithm which has complete knowledge of past and future, plus the ability to modify optimization variables in the past, so as to provide primal variables of the form $\mathbf{x}_t^\star \in \text{argmax}_{\mathbf{x}\in\mathcal{X}_t}\mathcal{L}_t(\mathbf{x},\boldsymbol{\lambda}^\star)$ for each round t given the offline dual optimal variables $\boldsymbol{\lambda}^\star$. We expect at the outset that the above algorithm should achieve performance which is a function of the oracle algorithm's performance plus factors that depend on the number of rounds T, as well as variability of constraints as measured by the oracle's ability to satisfy constraints. We present the bound in Theorem 1, and direct the details of the proof (as well as key technical Lemmas) to [12]. We note that the version of the theorem which we state is for instance valid for the choice of $\mathcal{E}$ we make in the experimental section.

THEOREM 1. *Let $\{\mathbf{x}_t^\star\}_{t=1}^T$ be the offline primal optimal assignment variables, and consider the sequence $\{\hat{\mathbf{x}}_t\}_{t=1}^T$ generated by our online algorithm. Let $\boldsymbol{e}_t^\star=\mathbf{A}_t\mathbf{x}_t^\star-\mathbf{b}_t$ be the corresponding optimal constraint residuals. We denote by*

$$f^\star\equiv f(\mathbf{x}_1^\star,\cdots,\mathbf{x}_T^\star)\equiv\frac{1}{T}\sum_{t=1}^{T}\mathbf{u}_t^\top\mathbf{x}_t^\star-\mathcal{E}(\frac{1}{T}\sum_t\mathbf{A}_t\mathbf{x}_t^\star-\mathbf{b}_t)$$

the optimal primal objective value obtained for the assignments $\{\mathbf{x}_t^\star\}_{t=1}^T$, and define $\hat{f}\equiv f(\hat{\mathbf{x}}_1,\cdots,\hat{\mathbf{x}}_T)$ similarly. Provided that $\max_{t=1,\cdots,T,\mathbf{x}_t\in\mathcal{X}_t,\boldsymbol{\lambda}\in\Lambda}\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x}_t,\boldsymbol{\lambda})\|_2\leq G$ for $G>0$, the projected online gradient descent/subgradients updates for $\hat{\boldsymbol{\lambda}}_t$ and $\hat{\mathbf{A}}_t$ with step sizes $\eta_t=L/t,\nu_t=R_A/\sqrt{t}$ imply

$$f^\star-\hat{f}\leq\frac{GL}{T}(1+\log(T))\cdot\max_{t=1,\cdots,T}\left\|\sum_{j=1}^{t}\frac{T-t}{T}\mathbf{e}_j^\star-\sum_{j=t+1}^{T}\frac{t}{T}\mathbf{e}_j^\star\right\|_2$$
$$+\frac{G^2L}{2T}(1+\log(T))+\frac{6R_\lambda R_x}{\sqrt{T}}\left[\sum_{t=1}^{T}\|\mathbf{A}_t-\mathbf{A}_{t+1}\|_F+R_A\right]$$

where $\mathcal{E}$ has Lipschitz gradients with modulus L.

Theorem 1 establishes that the projected online gradient descent updates for dual variables $\hat{\boldsymbol{\lambda}}_t$, along with projected online subgradient updates for constraint matrices $\hat{\mathbf{A}}_t$, yield an overall cumulative regret bound that decomposes into three terms. The first term measures the smoothness, or regularity, of the sequences $\{\mathbf{e}_t^\star\}_{t=1}^T$ (as measured by the oracle's ability to minimize constraint violations) as well as $\{\mathbf{u}_t\}_{t=1}^T$ (which have a direct influence on $\mathbf{x}_t^\star$). We can observe that if the residual vectors are constant $\mathbf{e}_t^\star=\mathbf{e}^\star$ for all t, the first term vanishes.[3] The second term captures the contribution of having to perform online updates, which is is sub-linear in the number of rounds T given previous results in the online convex optimization literature [11, 19]). The third and last term captures the contribution to the regret due to the variation in constraint matrices $\mathbf{A}_t$ as measured by $\sum_{t=1}^T\|\mathbf{A}_t-\mathbf{A}_{t+1}\|_F$.

We note that Theorem 1 does not require any assumptions about the arrival order of bid requests, stationarity, or about the precise mechanisms under which feedback is generated (e.g., specific auction dynamics, marketplace characteristics, etc). Thus, Theorem 1 holds for *any* sequences of $\{\mathbf{A}_t,\mathbf{b}_t\}_{t=1}^T,\{\mathbf{u}_t\}_{t=1}^T$, including those generated by an adversary, with the consequence that if the adversary is sufficiently powerful such that $\{\mathbf{A}_t\mathbf{x}_t^\star-\mathbf{b}_t\}_{t=1}^T,\{\mathbf{u}_t\}_{t=1}^T$ vary wildly (i.e., even the oracle algorithm is unable to obtain a smooth sequence $\{\mathbf{A}_t\mathbf{x}_t^\star-\mathbf{b}_t\}_{t=1}^T$), then the worst-case bound on regret for our proposed online primal-dual method can be large. Conversely, if the oracle algorithm is able to achieve a regular-enough sequence $\{\mathbf{A}_t\mathbf{x}_t^\star-\mathbf{b}_t\}_{t=1}^T$, then our proposed online primal-dual method is able to achieve the same performance plus a factor that is sublinear in the number of rounds.

5. DISTRIBUTED ASPECTS OF ONLINE AD ALLOCATION

Having presented regret bounds for our proposed online algorithm, we now address the distributed aspects of ad allocation, which are a fundamental to the display advertising setting in practice. We note that for the allocation problem, we can write $\mathbf{u}_t^\top\mathbf{x}_t=\sum_{h=1}^H\mathbf{u}_{ht}^\top\mathbf{x}_{ht}$ so that the primal step of computing $\hat{\mathbf{x}}_t\in\text{argmax}_{\mathbf{x}_t\in\mathcal{X}_t}\mathcal{L}_t(\mathbf{x}_t,\hat{\boldsymbol{\lambda}}_t)$ can be decomposed across multiple hosts, each processing disjoint subsets of bid requests for round t by ranking candidate ads for each bid request to be processed on that host. We first present an offline distributed algorithm that leverages this decomposability, which we will quickly incorporate into an online and distributed algorithm.

5.1 Offline dual decomposition

If we temporarily ignore the online aspect of the ad allocation problem, a technique that would allow us to perform distributed optimization is to decompose the ad allocation problem across hosts for a given set of dual variables, collect primal solutions and then update the dual variables. We note that for any round t, we can partition the primal variables $\mathbf{x}_t$ across hosts into disjoint sub-vectors of variables such that $\mathbf{x}_t=[\mathbf{x}_{1t},\cdots,\mathbf{x}_{Ht}]$, where $\mathbf{x}_{ht}$ is the vector of primal variables associated with host $h=1,\cdots,H$. Furthermore, by defining partitions I_h of bid requests across hosts $h=1,\cdots,H$, we can perform similar partitions for $\mathbf{u}_t,\mathbf{A}_t$, $\mathcal{X}_t$ (so that $\mathbf{x}_{ht}\in\mathcal{X}_{ht}$). With these partitions defined, the Lagrangian $\mathcal{L}_t(\mathbf{x},\boldsymbol{\lambda})$ can then be decomposed as

$$\begin{aligned}\mathcal{L}_t(\mathbf{x}_t,\boldsymbol{\lambda})&=\mathbf{u}_t^\top\mathbf{x}_t-\boldsymbol{\lambda}^\top\Big(\mathbf{A}_t\mathbf{x}_t-\mathbf{b}_t\Big)+\mathcal{E}^\star(\boldsymbol{\lambda})\\&=\sum_{h=1}^{H}\mathbf{u}_{ht}^\top\mathbf{x}_{ht}-\boldsymbol{\lambda}^\top\mathbf{A}_{ht}\mathbf{x}_{ht}+\boldsymbol{\lambda}^\top\mathbf{b}_t+\mathcal{E}^\star(\boldsymbol{\lambda})\\&=\sum_{h=1}^{H}F_{ht}(\mathbf{x}_{ht},\boldsymbol{\lambda})+G_t(\boldsymbol{\lambda}),\end{aligned}$$

[3] We refer the reader to [12] for an in-depth discussion.

which is a sum over functions of $\mathbf{x}_{ht}$ with no overlap in primal variable arguments, plus a function of $\boldsymbol{\lambda}$ that is otherwise independent of variables $\mathbf{x}_{ht}$. This implies that we can parallelize the primal step of estimating the allocation $\hat{\mathbf{x}}_t \in \arg\max_{\mathbf{x}_t \in \mathcal{X}_t} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})$ across a number of hosts for a given $\boldsymbol{\lambda}$, collect the solutions $\hat{\mathbf{x}}_{ht}$ and then update $\boldsymbol{\lambda}$. The resulting distributed primal-dual algorithm is then

- (P-offline & distributed) Compute for each round $t = 1, \cdots, T$ and for all hosts $h = 1, \cdots, H$:

$$\hat{\mathbf{x}}_{ht} \in \arg\max_{\mathbf{x}_{ht} \in \mathcal{X}_{ht}} F_{ht}(\mathbf{x}_{ht}, \boldsymbol{\lambda}),$$

- (D-offline) Collect solutions to form $\hat{\mathbf{x}}_t = [\hat{\mathbf{x}}_{1t}, \cdots, \hat{\mathbf{x}}_{Ht}]$ for all $t = 1, \cdots, T$. From $\hat{\mathbf{x}}_t$, compute $\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda})$ and gradient $\nabla\mathcal{L}(\hat{\mathbf{X}}, \boldsymbol{\lambda}) = \sum_t \nabla\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda})$. Given the gradient, update dual variables $\boldsymbol{\lambda}$ using a projected gradient descent method with step size η.

The proposed algorithm consists of iteratively 1) performing a distributed Lagrangian maximization across all hosts to obtain $\max_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$, followed by 2) a gradient descent method. This technique for solving large-scale convex optimization problems in a distributed setting consists of *dual decomposition* [7, 8, 13], which yields primal and dual-optimal $\mathbf{x}^\star, \boldsymbol{\lambda}^\star$ for a large enough number of iterations. However, in the online advertising setting, we do not have the ability to revise past ad allocations, nor can we realistically anticipate (and hence make allocation decisions for) future bid request opportunities. To address this whilst retaining the ability to decompose the problem across a large number of ad serving hosts, we next present online dual decomposition (ODD), an extension of the dual decomposition technique to the online optimization setting.

5.2 Online dual decomposition (ODD)

Consider now the following distributed and online algorithm for each round $t = 1, \cdots, T$:

- (P-online & distributed) Define

$$\hat{F}_{ht}(\mathbf{x}_{ht}, \boldsymbol{\lambda}) = \mathbf{u}_{ht}^\top \mathbf{x}_{ht} - \boldsymbol{\lambda}^\top \hat{\mathbf{A}}_{ht} \mathbf{x}_{ht}.$$

  Compute for all hosts $h = 1, \cdots, H$:

$$\hat{\mathbf{x}}_{ht} = \arg\max_{\mathbf{x}_{ht} \in \mathcal{X}_{ht}} \hat{F}_{ht}(\mathbf{x}_{ht}, \hat{\boldsymbol{\lambda}}_t).$$

- (D-online) Collect solutions to form $\hat{\mathbf{x}}_t = [\hat{\mathbf{x}}_{1t}, \cdots, \hat{\mathbf{x}}_{Ht}]$. From $\hat{\mathbf{x}}_t$ and given $\mathbf{A}_t$, compute $\mathcal{L}_t(\hat{\mathbf{x}}_t, \hat{\boldsymbol{\lambda}}_t)$ and gradient $\nabla\mathcal{L}_t(\hat{\mathbf{x}}_t, \hat{\boldsymbol{\lambda}}_t)$, and update dual variables $\hat{\boldsymbol{\lambda}}_{t+1}$ using a projected gradient descent method with step size η_t.
- (A-online) Given $\mathbf{A}_t, \hat{\mathbf{A}}_t$, update $\hat{\mathbf{A}}_{t+1}$ via a projected online subgradient method with step size ν_t.

The above distributed primal step, combined with the online projected gradient descent for $\hat{\boldsymbol{\lambda}}_t$ and subgradients for $\hat{\mathbf{A}}_t$ together form online dual decomposition (ODD), which extends classical dual decomposition to an online setting in which we must make ad allocations and update dual variables in an online fashion whilst having to use estimates of constraint matrices $\hat{\mathbf{A}}_t$. In ODD, in each round t we 1) solve for allocations $\hat{\mathbf{x}}_{ht}$ that maximize $\hat{F}_{ht}(\mathbf{x}_{ht}, \hat{\boldsymbol{\lambda}}_t)$ (distributed Lagrangian maximization with estimates of constraints $\hat{\mathbf{A}}_t$), followed by 2) updates of the dual variables $\hat{\boldsymbol{\lambda}}_t$ via an online dual gradient descent method on $\mathcal{L}_t(\hat{\mathbf{x}}_t, \hat{\boldsymbol{\lambda}}_t)$ and 3) online updates on $\hat{\mathbf{A}}_{t+1}$. We note that the above algorithm naturally leverages the embarassingly parallel and computationally-efficient ranking algorithm resulting from Lemma 1 for the primal step, and otherwise enjoys the same regret bound as presented in Theorem 1. The only additional assumption we've made for ODD is that there exists a centralized system for aggregating ad allocations, i.e., budget spend per round and per host given by $\sum_{i \in I_t \cap I_h} \mathbf{A}_{ht} \hat{\mathbf{x}}_{ht}$ and constraint matrices $\mathbf{A}_{ht}$ across hosts at the end of each round. From those pieces of information, global constraint violations can be computed for each ad (given centralized knowledge of $\mathbf{b}_t$ for each round) and updates for dual variables can be computed and then propagated back to each host in the ad serving fleet. Such systems are generally used in the online display advertising platforms to track budget spend: we do not discuss details on systems architectures that could be used for this purpose.

6. RESULTS

To validate our proposed online distributed ad allocation algorithm, we conducted three series of experiments with the aim to empirically compare the performance and delivery characteristics of ODD with those of other online algorithms. We focus on the application of max-performance, target delivery ad allocation described in Example 1, for which we consider two key metrics: 1) *performance* as measured by return on ad spend (ROAS), or the ratio of ad-attributed sales to ad spend, and 2) *delivery*, as measured by the fraction of an ad's budget that was successfully consumed relative to the fraction of the ad's flight time that has elapsed. More specifically, the goal in this class of problems is to maximize the ad's performance whilst consuming as close as 100% of the ad's budget by the end of its run time. We chose $\mathcal{E}(\mathbf{z})$ to be the Huber function (see the detailed discussion in [12]) for some non-negative parameters R, L:

$$\mathcal{E}(\mathbf{z}) = \begin{cases} \frac{L}{2}\|\mathbf{z}\|_2^2 & \text{if } \|\mathbf{z}\|_2 \leq \frac{R}{L}, \\ R\|\mathbf{z}\|_2 - \frac{R^2}{2L} & \text{otherwise.} \end{cases}$$

This choice yields the strongly convex dual function $\mathcal{E}^\star(\boldsymbol{\lambda}) = \mathcal{I}_{\mathcal{B}_R}(\boldsymbol{\lambda}) + \frac{1}{2L}\|\boldsymbol{\lambda}\|_2^2$, where $\mathcal{I}_{\mathcal{B}_R}$ is the indicator on the ℓ_2 ball such that $\|\hat{\boldsymbol{\lambda}}_t\|_2 \leq R$ for all rounds.

6.1 Experiment A: Simulation of ad allocation

In this experiment, we simulated ad allocation for a sample of 36 ads configured with a variety of budgets, delivery profiles (i.e., the fraction of budget to be delivered in a given time period) and targeting constraints encountered in our ad serving system, where ad impressions were allocated over a period of 24 hours with over 100MM bid request opportunities. To compare the performance and delivery characteristics of ODD, we examined 1) the proportional control heuristic of [6, 15] and 2) the *pdAvg* approximation algorithm from [10]. The proportional control heuristic consists of modifying the welfare term w_{ij} for each ad by subtracting a term that is proportional to the error in delivery in a given round (the difference between expected budget to be consumed in a round and actual amount delivered). The *pdAvg* approximation algorithm consists of modifying the welfare in similar fashion, but as a function of average welfare gained from allocating impressions for a given ad so far. All algorithmic parameters for the above algorithms were chosen from a discrete set in order to minimize total under- and over-delivery.

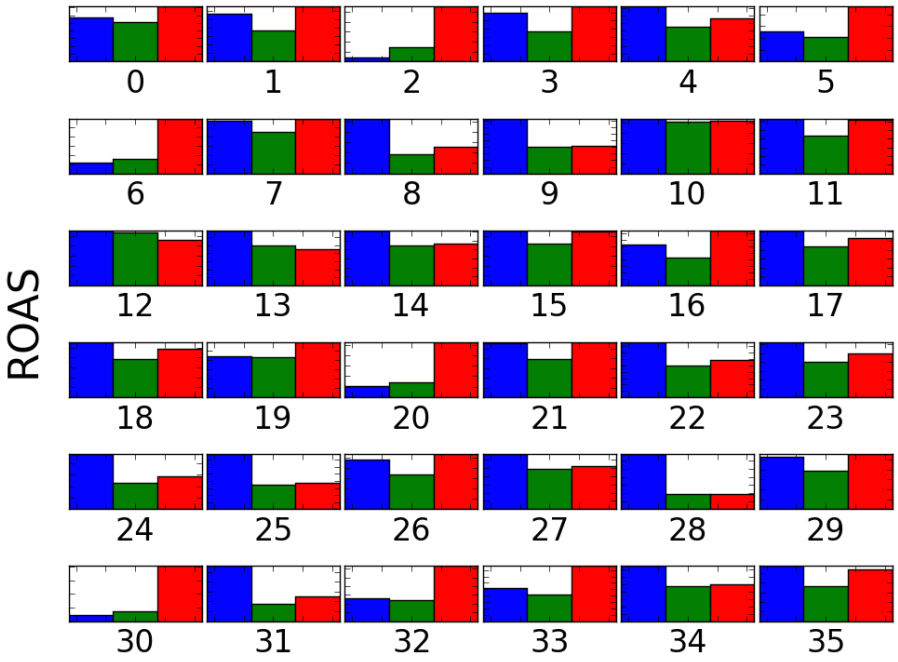


Figure 2: Performance for each of the 36 ads in Experiment A, as measured by return on ad spend (ROAS) obtained by ODD (blue), the proportional control heuristic (green) and the *pdAvg* algorithm of [10] (red). Subplots are numbered by ad to ease comparison between Figures 2 and 3. Figure best seen in color.

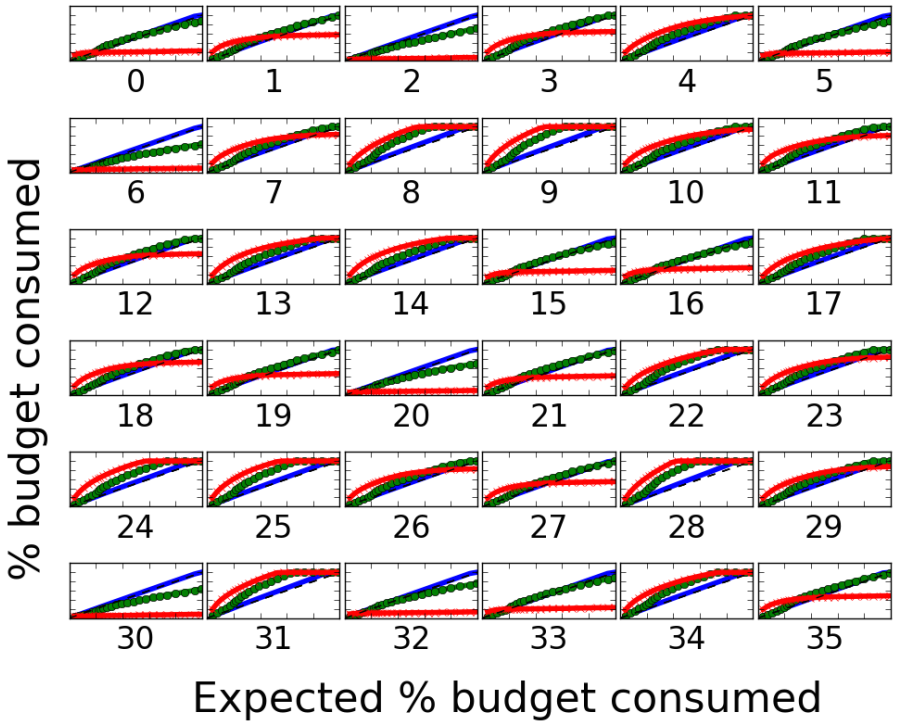


Figure 3: Percent budget delivered as a function of expected percentage of budget consumed over a period of 24 hours for the 36 ads for ODD (blue), the proportional control heuristic (green circles) and the *pdAvg* algorithm of [10] (red) for Experiment A. The black dotted line denotes the delivery profile matching exactly with the expected delivery profile for a given ad. Subplots are numbered by ad to ease comparison between Figures 2 and 3. Figure best seen in color.

Figure 2 shows the performance (ROAS) metrics at the end of the flight time for each of the three algorithms. Figure 3 shows delivery profiles for each one of them as a function of the expected delivery. As can be seen, *pdAvg* is able to achieve improved performance over ODD (for 16/36 ads), but at the cost of under-delivery in such cases (e.g., see ads 2, 5 and 30). This is undesirable from the advertiser's perspective, as under-delivery limits the overall impact of an ad and does not fully utilize the ad's budget to achieve the advertiser's performance goals. Conversely, for cases where *pdAvg* over-delivered (i.e., spent more budget ahead of schedule), such as for ads 25 and 31, the performance obtained was lower than that achieved by ODD. In aggregate, the performance metrics obtained for the ODD and *pdAvg* algorithms were very similar (within 3% of one another), but *pdAvg* demonstrated the highest variability in delivery (consistent with observations in [3]) in the form of over- or under-delivery (standard deviation in percent budget consumed across the 36 ads was 0.16% for ODD, 11.11% for proportional control, 34.2% for *pdAvg*) By comparison, the proportional control heuristic was only able to outperform the ODD technique for 4/36 ads (in all four cases, it under-delivered relative to the expected delivery profile), whilst yielding larger violations of delivery goals across the 36 ads as compared to ODD, but less variability in delivery as compared to *pdAvg*. Overall, ODD yielded a smoother delivery profile for all 36 ads with significantly less over- or under-delivery as compared to the other algorithms, and with performance (reward) equal to that achieved by the best of the algorithms studied.

6.2 Live experiments in a distributed production system at Amazon

The next two experiments were conducted within the distributed ad serving system at Amazon, where we compare the performance and delivery characteristics of ODD relative to a variant of the *pdAvg* algorithm.

6.2.1 Experiment B: Online A/B test at medium-scale

For the second experiment, we ran online tests of the ODD framework in a live production environment at Amazon where we compare our distributed optimization algorithm (treatment) to an ad allocation algorithm derived from *pdAvg* (control). Figure 4 shows results from an online test consisting of 7 pairs of control and treatment ads with a variety of slot positions, slot sizes and use/non-use of targeting, with the ad line items in treatment and control set to be otherwise identical. For this test, the exposure to external and non-stationary ad marketplace dynamics over the duration of the experiment (14 days) can lead to under- and over-delivery (where in practice, advertisers are less sensitive to over-delivery than under-delivery). As can be seen, the use of ODD achieves positive performance lifts, with delivery close to the target of 100% delivery for each ad.

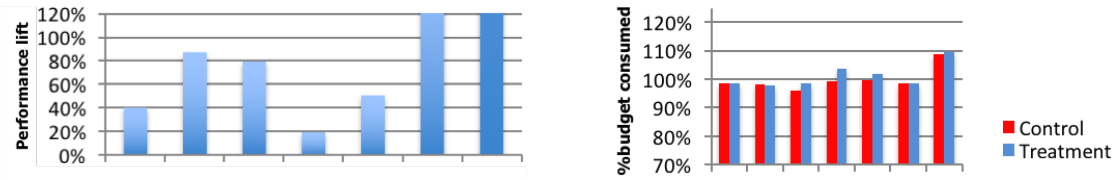
6.2.2 Experiment C: Online A/B test at large-scale

Figures 5 and 6 show the results for the last experiment, where we compare our distributed optimization algorithm (treatment) to an ad allocation algorithm derived from *pdAvg* (control). It is conducted with a larger-scale test (>25X the number of ads from Experiment B) than the previous experiment (leveraging a variety of slot positions, slot sizes and use/non-use of targeting) for a larger set of ads across 8 advertiser pools over a period of 21 days: we see a consistent and significant reduction in both over- and under-delivery for all ads in the treatment set. We also note an overall significant decrease (i.e., 39%) in the number of ads in the treatment group that under-delivered as compared to the control group, and a 22% decrease in the number of ads in the treatment group that over-delivered as compared to the control group, where under- and over-delivery were defined by each advertiser. We also observe an overall 9% decrease in the number of ads that under-perform and a 19% increase in the number of ads that achieve advertiser-specified performance targets (also defined by the advertiser). Last but not least, we observed overall performance lifts in the range of 10% to 193%: these results, taken cumulatively, demonstrate that our algorithm is able to achieve higher advertiser performance whilst being able to better enforce ad delivery constraints as compared to the control treatment.

7. DISCUSSION

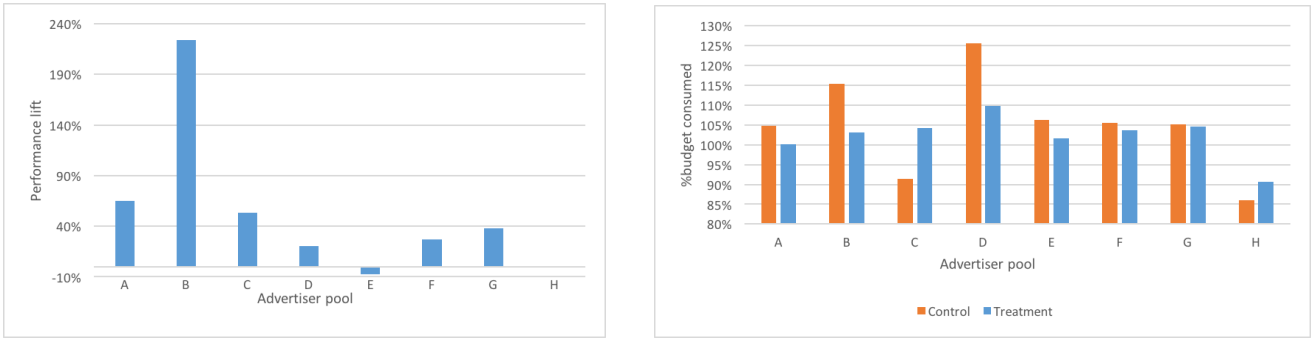
In this paper we have presented online dual decompositions as a technique for deriving practical online and distributed ad allocation algorithms for a variety of problems in online display advertising. In addition to having provided several empirical results derived from online advertising data, we have provided a theoretical analysis of the dynamic regret incurred by ODD, which includes a dependence on 1) the ability of an oracle algorithm to smoothly deliver ad impressions and 2) the smoothness with which constraints and constraint violations occur. While our analysis provides a regret bound that otherwise scales sub-linearly with the number of rounds T, there remain several factors that arise in practice that can impact performance and delivery in a production setting outside of the allocation algorithm: examples of these include 1) changes in auction dynamics, 2) changes, misallocations of ad budgets and 3) mis-specification of other campaign-specific parameters, to name a few. The impact of such changes on delivery and performance in practice is not to be discounted (and such impacts captured by our analysis in impacting the ability of an oracle algorithm to achieve smooth delivery) and should be studied as part of future work.

In this paper we have provided regret analysis that does not require making any assumptions about how feedback is generated by the external ad marketplace (e.g., auction dynamics, stochastic versus adversarial pricing). While this provides a robust baseline for performance and delivery characteristics that can be expected from ODD in a practical setting, this does not preclude the use of stable domain knowledge and/or assumptions about the ad marketplace (e.g., forecasting data) that could be useful in practice in improving performance and delivery properties for ODD. We also did not analyze the impact of delays and delayed feedback on the algorithm: we conjecture that adaptive gradient methods [1, 16] for dealing with delays may be appropriate for this setting, which we leave for future work. In terms of dealing with incomplete feedback, we leave as future work an investigation into extensions of ODD into the bandits setting [5]. Last but not least, we did not examine different distributed systems architectures and their impact on both empirical performance and delivery of our framework, which would be fruitful directions for future investigation.



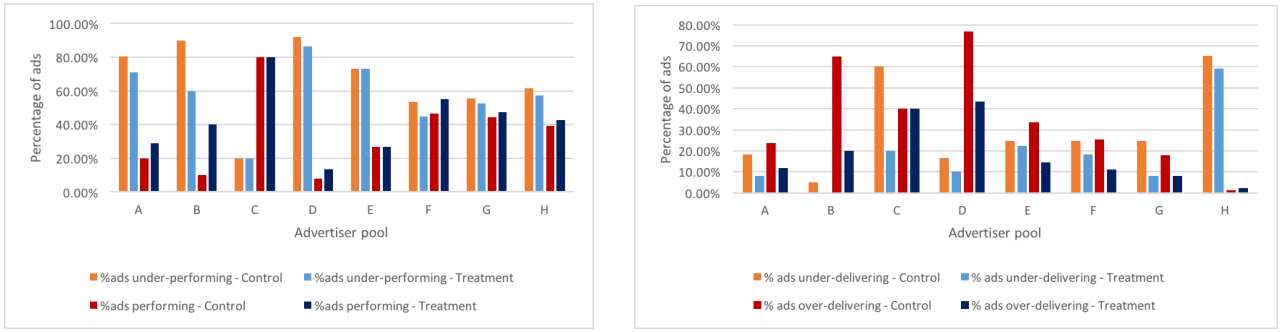
(a) (b)

Figure 4: (a) Performance lift for treatment ads and (b) change in delivery for treatment ads using ODD relative to control ads as a result of ad allocation via ODD in Experiment B. For this test, the exposure to external and non-stationary ad marketplace dynamics over the duration of the experiment (14 days) can lead to under- and over-delivery (in practice, advertisers are less sensitive to over-delivery than under-delivery). Here, the use of ODD achieves positive performance lifts, with delivery close to the target of 100% delivery for each ad. Figure best seen in color.



(a) (b)

Figure 5: (a) Performance and (b) delivery characteristics obtained from ODD relative to control ads for Experiment C. For this test, the exposure to external and non-stationary ad marketplace dynamics over the duration of the experiment (21 days) can lead to under- and over-delivery (where in practice, advertisers are less sensitive to over-delivery than under-delivery). Here we see overall performance lifts in the range of 10% to 193% and a consistent and significant reduction in both over- and under-delivery for all ads in the treatment set. Figure best seen in color.



(a) (b)

Figure 6: (a) Lift in performance and (b) change in delivery relative to control ads as a result of ad allocation via ODD for Experiment C. For this test, the exposure to external and non-stationary ad marketplace dynamics over the duration of the experiment (21 days) can lead to under- and over-delivery (in practice, advertisers are less sensitive to over-delivery than under-delivery). Here we see decreases in the fraction of ads that under-perform and/or under-deliver, as well as an increase in the fraction of ads that perform or deliver up to advertiser-specified expectations. Figure best seen in color.

8. REFERENCES

[1] A. Agarwal and J. C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 873–881, 2011.

[2] S. Agrawal and N. R. Devanur. Fast algorithms for online stochastic convex programming. In *SODA 2015 (ACM-SIAM Symposium on Discrete Algorithms)*. SIAM-Society for Industrial and Applied Mathematics, January 2015.

[3] A. Bhalgat, J. Feldman, and V. Mirrokni. Online allocation of display ads with smooth delivery. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2012)*, pages 1213–1221, 2012.

[4] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] O. Chapelle, E. Manavoglu, and R. Rosales. Simple and scalable response prediction for display advertising. *ACM Trans. Intell. Syst. Technol.*, 5(4):61:1–61:34, Dec. 2014.

[6] Y. Chen, P. Berkhin, B. Anderson, and N. Devanur. Online bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1307–1315, 2011.

[7] G. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[8] H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

[9] J. Feldman, M. Henzinger, N. Korula, V. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. In *ESA'10 Proceedings of the 18th annual European conference on Algorithms: Part I*, pages 182–194, 2010.

[10] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pal. Online ad assignment with free disposal. In *Proceedings of the 5th conference on Web and Internet Economics (WINE)*, 2009.

[11] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[12] R. Jenatton, J. Huang, C. Archambeau, and D. Csiba. Online optimization and regret guarantees for non-additive long-term constraints. *arXiv:1602.05394*, 2016.

[13] L. Lasdon. *Optimization theory for Large Systems*. MacMillan, 1970.

[14] C. Lemarechal and J.-B. Hiriart-Urruty. *Convex Analysis and Minimization Algorithms II*. Springer-Verlag, 1993.

[15] M. McEachran. Using Proportional Control For Better Pacing. `http://rubiconproject.com/technology-blog/using-proportional-control-for-better-pacing/`, 2012.

[16] B. McMahan and M. Streeter. Delay-tolerant algorithms for asynchronous distributed online learning. In *Advances in Neural Information Processing Systems*, pages 2915–2923, 2014.

[17] A. Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2012.

[18] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency, Volume A*. Springer, 2003.

[19] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

Appendix

PROOF (LEMMA 1). Note that maximizing $\mathbf{u}_t^\top \mathbf{x}_t$ with respect to $\mathbf{x}_t \in \mathcal{X}_t$ is itself an LP, which can be written as:

$$\begin{aligned} \underset{\mathbf{x}_t}{\text{maximize}} \quad & \sum_{i \in I_t, j \in C_i} (v_{ij} - w_i) x_{ij} \\ \text{subject to} \quad & \forall\, i \in I_t, \sum_{j \in C_i} x_{ij} \leq 1 \\ & \forall\, i \in I_t, j \in C_i,\ x_{ij} \geq 0. \end{aligned}$$

The dual of the above LP is given by

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \sum_{i \in I_t} \beta_i \\ \text{subject to} \quad & \forall\, i \in I_t, j \in C_i, \beta_i \geq v_{ij}, \\ & \forall\, i \in I_t, \beta_i \geq w_i, \end{aligned}$$

with dual optimum $\boldsymbol{\beta}^\star$. Trivially, for a given i, if $\beta_i^\star \geq w_i$, we must have $\beta_i^\star = \max_{j \in C_i} v_{ij} = v_{ij^\star}$ (otherwise $\beta_i^\star$ would violate the constraint $\beta_i^\star \geq v_{ij}\ \forall\, j \in C_i$, which contradicts dual optimality). Now, since a feasible solution to the above primal LP exists (trivially set all variables $x_{ij} = 0$) and the objective function is bounded, there exists at least one solution $\hat{\mathbf{x}}_t$ that maximizes $\mathbf{u}_t^\top \mathbf{x}_t$. Suppose that for a given i, we have $\beta_i^\star > 0$ and we were to assign $x_{ij'} = 1, x_{ij} = 0\ \forall\, j \neq j'$ for some $j' \neq j^\star$: call this vector of primal variables $\mathbf{x}'_t$. Then the primal objective here would be $\mathbf{u}_t^\top \mathbf{x}'_t < \mathbf{u}_t^\top \hat{\mathbf{x}}_t$, since $v_{ij^\star} > v_{ij'}$ (assuming no ties), which contradicts. Therefore $x_{ij^\star} = 1, x_{ij} = 0\ \forall\ j \neq j^\star$ maximizes $\mathbf{u}_t^\top \mathbf{x}'_t$. Finally, the $\hat{x}_{ij}$'s are integral 0/1 variables (i.e.: we cannot have fractional x_{ij}'s that maximize $\mathbf{u}_t^\top \mathbf{x}_t$: to see this, we note that the optimal allocation problem is equivalent to matching bid requests $i \in U$ to a single ad $j \in V$ in a bi-partite graph $G = (U, V, E)$ where U is the set of bid requests, V is the set of ads and E is the set of edges connecting nodes in U to nodes in V. In particular, for a given bid request $i \in U$, the set of edges incident on i is C_i. Now, the constraint in the primal LP can be written as $\mathbf{G}\mathbf{x} \leq \mathbf{1}$ with $\mathbf{G}$ being the incidence matrix of G. Since G is (by construction) bi-partite, $\mathbf{G}$ is totally unimodular [18] and so the primal LP only has integral 0/1 solutions. □