

Optimal Allocation of Real-Time-Bidding and Direct Campaigns

Grégoire Jauvion

AlephD

Paris

gregoire.jauvion@gmail.com

Nicolas Grislain

AlephD

Paris

ng@alephd.com

ABSTRACT

In this paper, we consider the problem of optimizing the revenue a web publisher gets through real-time bidding (i.e. from ads sold in real-time auctions) and direct (i.e. from ads sold through contracts agreed in advance). We consider a setting where the publisher is able to bid in the real-time bidding auction for each impression. If it wins the auction, it chooses a direct campaign to deliver and displays the corresponding ad.

This paper presents an algorithm to build an optimal strategy for the publisher to deliver its direct campaigns while maximizing its real-time bidding revenue. The optimal strategy gives a formula to determine the publisher bid as well as a way to choose the direct campaign being delivered if the publisher bidder wins the auction, depending on the impression characteristics.

The optimal strategy can be estimated on past auctions data. The algorithm scales with the number of campaigns and the size of the dataset. This is a very important feature, as in practice a publisher may have thousands of active direct campaigns at the same time and would like to estimate an optimal strategy on billions of auctions.

The algorithm is a key component of a system which is being developed, and which will be deployed on thousands of web publishers worldwide, helping them to serve efficiently billions of ads a day to hundreds of millions of visitors.

CCS CONCEPTS

• **Mathematics of computing** → **Probability and statistics; Continuous optimization**; • **Applied computing** → **Online auctions**;

KEYWORDS

High-Dimensional Optimization, Big-Data, Ad-tech, Real-time, Auctions

ACM Reference Format:

Grégoire Jauvion and Nicolas Grislain. 2018. Optimal Allocation of Real-Time-Bidding and Direct Campaigns. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19-23, 2018, London, United Kingdom*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3219819.3219877>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '18, August 19-23, 2018, London, United Kingdom

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5552-0/18/08...\$15.00

<https://doi.org/10.1145/3219819.3219877>

1 INTRODUCTION

Web publishers use different channels to sell their ads. We consider in this paper two channels: *RTB* (real-time bidding) and *Direct*. In *RTB*, the publisher sells ads through an ad exchange where buyers compete in auctions happening in real time. In *Direct*, the ads are sold through direct campaigns.

A direct campaign is a contract in which a publisher commits to deliver a specified number of ads to an advertiser while ensuring that some KPIs are met (for example the publisher may guarantee a minimum number of clicks to the advertiser). Should the publisher not deliver the impressions as defined in the contract, it pays a penalty to the advertiser.

A natural way for a publisher to allocate impressions between the two channels is to bid in the *RTB* auction for every impression: if it wins the auction, the publisher chooses a direct campaign to sell the impression, otherwise the impression is sold in the *RTB* market.

The first section gives a review of past work related to this topic. The second section states the optimization problem to solve. The two following sections derive optimality conditions and present an algorithm to build an optimal delivering strategy. The last section presents the results of experiments realized on a real *RTB* auctions dataset.

2 RELATED WORK

The problem of maximizing a web publisher advertising revenue is much studied in the literature. Because of the complexity of the online advertising ecosystem, and because the publishers sell their ads through different channels, this problem leads to diverse approaches.

[12], [15] and [3] consider a publisher selling its ads through *RTB* only, and define strategies for the publisher to set appropriate reserve prices in each auction to optimize its *RTB* revenue. In [12], the reserve prices are set in a static way (i.e. an optimal reserve price is estimated on a large set of auctions), whereas [15] and [3] define methodologies to set dynamic reserve prices which are predicted in real time before an auction happens.

[11] and [10] study the optimal delivery of a budget across time in the *RTB* market. This problem known as budget pacing has been much studied and presents strong similarities with the problem of delivering a direct campaign. In both articles, the bids are updated online to optimize the budget delivery.

[13] and [2] focus on *Direct* revenue maximization. They consider that the impressions can be sold in direct campaigns only, and define strategies to allocate optimally a set of impressions between all campaigns in order to maximize the publisher revenue (i.e. minimize the penalty the publisher pays if some campaigns are not delivered as agreed in the contract). [13] formulates this problem as a dynamic

programming problem, and [2] proposes an efficient algorithm to allocate near-optimally the impressions between the campaigns.

[7], [6], [5] and [1] study the joint optimization of the publisher revenue coming from *RTB* and *Direct*. They consider a setting similar to our setting, in which the publisher bids in auctions to compete with *RTB* bidders.

In [7], the quality of a direct campaign delivery is represented by a utility function, and the quality of each impression is linked directly to its price in the *RTB* market, whereas we characterize a campaign delivery by some KPIs.

The model used in [6] and [5] is based on the modeling of the supply and the demand on an aggregated level, while we analyze data auction per auction without needing to model the supply and demand.

In [1], a campaign is defined by a fixed number of impressions which is supposed to be delivered exactly, and a per-impression quality. The algorithm aims to maximize a weighted sum of the *RTB* revenue and of the direct campaigns qualities. The main difference with our approach is that the campaigns are supposed to be completely delivered. In our approach, we enable under-delivery of a campaign if it brings a higher revenue to the publisher, which makes the problem more complex. Also, the approach described in [1] may be hard to scale due to the allocation plan used to allocate similar campaigns. We solve this issue by regularizing the optimization problem.

Finally, [14] and [4] present stochastic models for supply and demand, and link the problem of *Direct* revenue maximization with the problem of option pricing in finance.

3 PROBLEM STATEMENT

In this section, we introduce the two channels used by the publisher to sell its ads: *RTB* and *Direct*. Then, we derive the global revenue the publisher gets from both channels, and formulate the revenue optimization problem.

All random variables are defined with respect to the filtration \mathcal{F}_{t_0} , which is the filtration representing the information available at time $t = t_0$, i.e. before the first auction happens.

3.1 RTB market

We represent the *RTB* market as a set of N time-ordered auctions happening at times $t_1, \dots, t_n, \dots, t_N$. This approach is deterministic in that we do not model the uncertainty in the auctions to come.

In auction n , a set of *RTB* buyers compete to buy the ad slot and display their ad. We note respectively B_n and C_n the random variables corresponding to the highest bid and the second highest bid in the auction (possibly 0 if there are not enough buyers bidding in the auction).

We note $F_n(\cdot)$ (respectively $f_n(\cdot)$) the marginal cumulative distribution function of B_n (respectively its probability density function).

3.2 Direct campaigns

The *Direct* channel is represented by K direct campaigns C_1, \dots, C_K booked by the publisher at time $t = t_0$. We assume that the set of campaigns is static, i.e. existing campaigns can not be cancelled, and no new campaign is booked.

Each campaign is defined by a contractual revenue, which is the revenue the publisher is paid to deliver the campaign, a set of goals and a penalty function giving the penalty the publisher has to pay when these goals are not met.

3.2.1 Contractual revenue of the campaign. We note π_k the revenue the publisher gets to deliver the campaign. It is worth noting that the publisher strategy to deliver the campaign does not depend on this amount, which is contractual and fixed before the campaign is delivered.

3.2.2 Goals of the campaign. Let p_k be the number of goals for campaign C_k . Each goal i is defined by a targeting $\mathcal{T}_{k,i}$ and a volume $g_{k,i}$ to be delivered in the targeting $\mathcal{T}_{k,i}$.

The targeting $\mathcal{T}_{k,i}$ is defined in each auction n by the Bernoulli variable $X_{n,k,i}$ equal to 1 if impression n is eligible for the targeting, and 0 otherwise. Let $\theta_{n,k,i}$ be the parameter of the Bernoulli variable $X_{n,k,i}$, which is supposed to be known before the auction happens.

A deterministic targeting corresponds to the case where we know certainly if the ad will be in the targeting when bidding in the auction, and we have in this case $\theta_{n,k,i} = 1$ or $\theta_{n,k,i} = 0$. For example, targeting a website or an ad placement is a deterministic targeting.

A stochastic targeting corresponds to the case where we do not know if the ad will be in the targeting before the auction happens. In this case, $\theta_{n,k,i}$ is an estimation of the probability for the ad to be in the targeting. For example, targeting only clicked impressions is a stochastic targeting, as we do not know if the ad will be clicked or not before it has been displayed.

3.2.3 Penalty function. We introduce the variables $\delta_{k,1}, \dots, \delta_{k,p_k}$, where $\delta_{k,i}$ corresponds to the difference between the goal $g_{k,i}$ and the volume delivered effectively on the targeting $\mathcal{T}_{k,i}$.

For campaign C_k , the penalty function $L_k(\delta_{k,1}, \dots, \delta_{k,p_k})$ gives the penalty the publisher has to pay when it fails to deliver the goals. At the end of the campaign, this penalty should be deduced from the contractual revenue of the campaign to get the revenue the publisher got from the campaign.

A typical penalty function would be $L_k(\delta_{k,1}, \dots, \delta_{k,p_k}) = \sum_{i=1}^{p_k} l_i \times \max(\delta_{k,i}, 0)$, corresponding to the case where the publisher pays l_i for each non-delivered impression on targeting $\mathcal{T}_{k,i}$.

3.3 Auction mechanics

Let a_n and $R_n(a_n)$ be respectively the publisher bid and the *RTB* revenue in auction n . We note also $r_n(a_n) = \mathbb{E}[R_n(a_n)]$ the expected *RTB* revenue. The auction mechanics is the following:

- If $a_n \geq B_n$, the publisher bidder wins the auction and the publisher chooses a direct campaign to display. We note Q_n the multinoulli variable with parameters $(q_{n,1}, \dots, q_{n,K})$ such that campaign C_k is chosen with probability $q_{n,k}$. The *RTB* revenue for this auction is $R_n(a_n) = 0$
- If $a_n < B_n$, the ad is sold in the *RTB* market. When the auction mechanism is a second-price auction, which is the most usual mechanism, we have $R_n(a_n) = \mathbb{1}_{a_n < B_n} \max(C_n, a_n)^1$.

¹This outcome corresponds to the case where no reserve price is set in the auction. The case where a reserve price is set is handled by replacing C_n by the maximum value between the second highest bid and the reserve price.

When the auction is a first-price auction, we have $R_n(a_n) = \mathbb{1}_{a_n < B_n} B_n$

3.4 Volumes delivered on direct campaigns

Let $V_{k,i}$ be the volume delivered on the targeting $\mathcal{T}_{k,i}$ after the N auctions have occurred. It can be written:

$$V_{k,i} = \sum_{n=1}^N \mathbb{1}_{a_n \geq B_n} \mathbb{1}_{Q_n = k} X_{n,k,i}$$

We note $v_{k,i} = \mathbb{E}[V_{k,i}]$. We assume that for each (k,i) , the random variables B_n , Q_n and $X_{n,k,i}$ are independent, which gives:

$$v_{k,i} = \sum_{n=1}^N F_n(a_n) q_{n,k} \theta_{n,k,i}$$

Note that the independence hypothesis between B_n , Q_n and $X_{n,k,i}$ is not a strong hypothesis: indeed, as these random variables are specific to auction n , they include already all the information we may know about the auction.

3.5 Publisher revenue

Let Y be the total publisher revenue coming both from *RTB* and *Direct*, and let $y = \mathbb{E}[Y]$ be its expected value. It depends on the publisher bids $(a_n)_{n=1}^N$, and on the parameters $(q_{n,1}, \dots, q_{n,K})_{n=1}^N$ used to select the winning campaign when the publisher bidder wins the auction.

It writes:

$$Y \left((a_n)_{n=1}^N, (q_{n,1}, \dots, q_{n,K})_{n=1}^N \right) = \sum_{n=1}^N R_n(a_n) + \sum_{k=1}^K \left(\pi_k - L_k(V_{k,1} - g_{k,1}, \dots, V_{k,p_k} - g_{k,p_k}) \right) \quad (1)$$

To simplify the expression of the expected revenue, we make the following approximation:

$$\mathbb{E}[L_k(V_{k,1} - g_{k,1}, \dots, V_{k,p_k} - g_{k,p_k})] \approx L_k(\mathbb{E}[V_{k,1} - g_{k,1}], \dots, \mathbb{E}[V_{k,p_k} - g_{k,p_k}]) \quad (2)$$

This approximation tends to be well verified when the volumes delivered on all targetings are large.

The expected publisher revenue is finally written:

$$y \left((a_n)_{n=1}^N, (q_{n,1}, \dots, q_{n,K})_{n=1}^N \right) = \sum_{n=1}^N r_n(a_n) + \sum_{k=1}^K \left(\pi_k - L_k(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k}) \right) \quad (3)$$

Note that the publisher bidder impacts the *RTB* revenue in two ways: by removing impressions from the *RTB* channel to deliver direct campaigns, and by modifying the *RTB* closing price (for example when the publisher bid a_n acts as a second price in a second-price auction mechanism).

3.6 Optimization problem statement

In this section, we note (a_n) the sequence of all a_n ($n = 1, \dots, N$), $(q_{n,k})$ the sequence of all $q_{n,k}$ ($n = 1, \dots, N$ and $k = 1, \dots, K$), and $(v_{k,i})$ the sequence of all $v_{k,i}$ ($k = 1, \dots, K$ and $i = 1, \dots, p_k$).

The optimization problem is written the following way:

$$\max_{(a_n), (q_{n,k}), (v_{k,i})} \sum_{n=1}^N r_n(a_n) + \quad (4)$$

$$\sum_{k=1}^K \left(\pi_k - L_k(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k}) \right)$$

s.t

$$\begin{cases} \forall k, i \ v_{k,i} = \sum_{n=1}^N F_n(a_n) q_{n,k} \theta_{n,k,i} \\ \forall n \ \sum_{k=1}^K q_{n,k} = 1 \\ \forall n, k \ q_{n,k} \geq 0 \end{cases} \quad (5)$$

Note that the variables $(v_{k,i})$, which represent the volumes delivered on all targetings, are considered to be variables to optimize whereas they derive directly from (a_n) and $(q_{n,k})$. This relation is introduced as an equality constraint. Stating the problem this way makes its analysis and the presentation of the optimization algorithm much simpler.

4 OPTIMALITY CONDITIONS

This section writes the Karush-Kuhn-Tucker conditions of the optimization problem stated in the previous section for two kinds of penalty functions: differentiable functions and rectified linear functions (which are not differentiable everywhere but which are widely used in practice). The lagrangian of the problem and the complete derivation of the Karush-Kuhn-Tucker conditions are given in the appendix.

4.1 Optimality conditions when the penalty functions are differentiable

We introduce the score of campaign C_k in auction n , defined by:

$$c_{n,k} = - \sum_{i=1}^{p_k} \theta_{n,k,i} \frac{\partial L_k}{\partial v_{k,i}}(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k})$$

4.1.1 Optimality condition on $q_{n,k}$. The optimality condition on $q_{n,k}$ writes:

$$q_{n,k} > 0 \Rightarrow k \in \underset{k}{\operatorname{argmax}} c_{n,k}$$

In the particular case where there is one unique campaign C_{k_0} maximizing $c_{n,k}$, we have $q_{n,k} = 1$ if $k = k_0$ and $q_{n,k} = 0$ otherwise. If several campaigns maximize $c_{n,k}$, the optimality conditions do not give an easy way to determine the corresponding $q_{n,k}$.

4.1.2 Optimality condition on a_n . The optimality condition on a_n writes:

$$r'_n(a_n) + f_n(a_n) \max_k c_{n,k} = 0$$

It is worth noting that the expected revenue from auction n , should it bring c if it is sold in a direct campaign, writes $r_n(a) + F_n(a)c$. Therefore, a_n is the bid maximizing the expected revenue from auction n if it brings $\max_k c_{n,k}$ if it is sold in a direct campaign.

The appendix gives a detailed analysis of the existence of a_n verifying this condition.

4.2 Optimality conditions when some penalty functions are rectified linear functions

We assume in this section that the penalty function for campaign C_k writes $L_k(v_{k,1}, \dots, v_{k,p_k}) = \sum_{i=1}^{p_k} l_i \max(g_{k,i} - v_{k,i}, 0)$. This kind of penalty function is widely used in practice but is not differentiable.

The optimality conditions are similar to the ones obtained when the penalty functions are differentiable, but where the score of campaign C_k is written the following way:

$$c_{n,k} = \sum_{i=1}^{p_k} \theta_{n,k,i} \kappa_{k,i} \text{ with } \begin{cases} \kappa_{k,i} = l_i & \text{if } v_{k,i} < g_{k,i} \\ \kappa_{k,i} \in [0, l_i] & \text{if } v_{k,i} = g_{k,i} \\ \kappa_{k,i} = 0 & \text{if } v_{k,i} > g_{k,i} \end{cases}$$

5 OPTIMIZATION ALGORITHM

This section proposes an algorithm to estimate an optimal strategy (i.e. (a_n) and $(q_{n,k})$) on a *RTB* auctions dataset. We assume that the dataset is processed in batches of auctions. In practice, the dataset is shuffled before processing it to ensure the batches are statistically equivalent.

5.1 Description of the algorithm when the penalty functions are differentiable

When the penalty functions are differentiable, the algorithm estimates the optimal volumes delivered on all targetings, which are the only variables needed to estimate in each auction n the scores of the campaigns and then the optimal a_n and $q_{n,k}$.

We note $v_{k,i}^{(j)}$ the estimation of these volumes after having processed j batches. A naive initialization of the algorithm is $v_{k,i}^{(0)} = 0$, but a better initialization can accelerate significantly the convergence of the algorithm. For example, it may be relevant to initialize the algorithm with $v_{k,i}^{(0)} = g_{k,i}$.

The j -th batch of auctions is processed with two steps.

5.1.1 Step 1: loop on the batch and apply the strategy determined using $v_{k,i}^{(j-1)}$. In this section, we note $(c_{n,k}^{(j-1)})$ the scores of the campaigns in auction n computed when the volumes delivered on all targetings are $(v_{k,i}^{(j-1)})$.

We note $\widehat{a_n}$ and $\widehat{q_{n,k}}$ the publisher bid and the campaigns probabilities applied in auction n . They are determined with the following equations, which are obtained by writing the optimality conditions when the volumes are given by $v_{k,i}^{(j-1)}$:

$$\begin{cases} \widehat{q_{n,k}} > 0 \Rightarrow k \in \arg\max_k c_{n,k}^{(j-1)} \\ r'_n(\widehat{a_n}) + f_n(\widehat{a_n}) \max_k c_{n,k}^{(j-1)} = 0 \end{cases} \quad (6)$$

When $\arg\max_k c_{n,k}^{(j-1)}$ contains one element only, it is selected, otherwise a campaign is randomly chosen among those with the maximum score. We note $\widehat{v_{k,i}}$ the volumes delivered on all campaigns when this strategy is applied on the j -th batch.

5.1.2 Step 2: determine $v_{k,i}^{(j)}$. We introduce the ratio of the number of auctions in one batch to the number of auctions in the dataset, noted ρ . We introduce also the learning rate of the algorithm α_j .

$v_{k,i}^{(j)}$ is determined with:

$$\forall k, i : v_{k,i}^{(j)} = v_{k,i}^{(j-1)} + \alpha_j (\widehat{v_{k,i}} - \rho \times v_{k,i}^{(j-1)})$$

This update methodology is based on the fact that at the optimum, the volume delivered on a targeting on one batch of data $\widehat{v_{k,i}}$ is equal in average to $\rho \times v_{k,i}^{(j-1)}$.

The choice of (α_j) has a major impact on the convergence speed of the algorithm. If α_j is too high, the algorithm may not converge, whereas a too low value makes the convergence very slow. We use in practice a decreasing α_j , for example $\alpha_j \propto \frac{1}{j}$.

The size of the batches is an important parameter of the algorithm. In practice, we notice that the smaller the batches, the faster converges the algorithm, where the convergence speed is defined as the number of auctions to process to attain a given level of convergence. However, the update step performed for each batch is computationally expensive as it requires to update the volumes for all campaigns. In practice, we use batches composed of a few hundreds of auctions.

5.2 Description of the algorithm when some penalty functions are rectified linear functions

We assume in this section that the penalty function for campaign C_k writes $L_k(v_{k,1}, \dots, v_{k,p_k}) = \sum_{i=1}^{p_k} l_i \max(g_{k,i} - v_{k,i}, 0)$. The score $c_{n,k}$ in auction n writes in this case $c_{n,k} = \sum_{i=1}^{p_k} \theta_{n,k,i} \kappa_{k,i}$.

For this campaign, the algorithm estimates the values of $\kappa_{k,i}$, which are the variables needed to estimate the score of this campaign in each auction n . We define $\kappa_{k,i}^{(j)}$ the estimate of $\kappa_{k,i}$ after having processed j batches.

A natural way to initialize the algorithm is $\kappa_{k,i}^{(0)} = 0$, which is consistent with the initialization $v_{k,i}^{(0)} = 0$. We could also initialize with $\kappa_{k,i}^{(0)} = l_i$.

Let $\widehat{\kappa_{k,i}}$ be the estimated value of $\kappa_{k,i}$ on the j -th batch. It is estimated by comparing the volume delivered on the batch and the volume we expect to be delivered if the goal $g_{k,i}$ is reached:

$$\widehat{\kappa_{k,i}} = \begin{cases} l_i & \text{if } \widehat{v_{k,i}} < \rho \times g_{k,i} \\ 0 & \text{otherwise} \end{cases}$$

Then, the update step is similar:

$$\forall i : \kappa_{k,i}^{(j)} = \kappa_{k,i}^{(j-1)} + \alpha_j (\widehat{\kappa_{k,i}} - \kappa_{k,i}^{(j-1)})$$

5.3 A regularization of the optimization problem

The application of the optimal strategy consists in picking in each auction the campaign with the highest score, where the score is computed using the volumes $(v_{k,i}^{(j)})$ after the algorithm has converged.

Such a choice of the winning campaign may lead to an unwanted behaviour of the algorithm. For example, let's consider the case of two campaigns with identical definitions. At the optimum, the volumes for these two campaigns should be identical, which means that in all auctions, both campaigns will be picked with equal probabilities. In practice, the optimal volumes estimated for these two

campaigns using the algorithm may be slightly different, and in this case it is easy to see that one of the campaigns will always have a higher score than the other one, which will consequently never be picked.

This simple example illustrates that a minor error in the estimation of the optimal volumes of the campaigns may have a large impact on the estimated optimal strategy.

To overcome this issue, we have written a regularized version of the optimization problem and derived the corresponding optimality conditions. The complete derivation is given in the appendix. The regularized optimality conditions are:

$$\begin{cases} q_{n,k} = \frac{\exp(c_{n,k} \frac{F_n(a_n)}{\rho})}{\sum_{k'=1}^K \exp(c_{n,k'} \frac{F_n(a_n)}{\rho})} \\ r'_n(a_n) + f_n(a_n) \sum_{k=1}^K q_{n,k} c_{n,k} = 0 \end{cases}$$

where ρ is the regularization parameter.

It is worth noting that using this regularization in the optimization phase may also accelerate the convergence of the optimization algorithm.

6 EXPERIMENTS ON AN AUCTIONS DATASET

We have used in this section a real *RTB* auctions dataset corresponding to one week of video advertising data. We have filtered auctions which led to an ad (i.e. auctions won by a buyer and for which a video ad has been delivered effectively). The dataset has been shuffled before running the optimization algorithm to ensure that the batches are statistically equivalent.

In each auction, we will use the following informations:

- The impression id, which is a unique identifier of the impression
- The placement id, which is a unique identifier of the ad slot. The dataset contains 4 different ad placements, called \mathcal{P}_1 , \mathcal{P}_2 , \mathcal{P}_3 and \mathcal{P}_4
- The highest bid of the auction (which is strictly positive because we filtered on auctions won by a buyer)
- We know if the impression has been viewed² or clicked

Two experiments have been performed:

- The first experiment runs the optimization algorithm with a few campaigns defined manually, and gives an interpretation of the optimal strategy
- The second experiment runs the optimization algorithm with many campaigns and shows that the algorithm scales with the number of campaigns

We assume that the auction mechanism is a first-price auction mechanism. This makes the optimality condition on a easier to solve because it does not need any hypothesis on the highest *RTB* bid distribution.

The optimization algorithm is run with batches of size 1000 and with $\alpha_t = \frac{1}{t}$. The optimality conditions have been regularized with the temperature parameter $\rho = 0.5$, both when determining the optimal strategy and when applying the optimal strategy on the dataset. The algorithm is initialized with $\kappa_{k,i} = 0$.

²The IAB viewability measure is used: an ad is considered to be viewed if at least 50% of its surface has been within the visible area of the browser window during at least 1 second.

We define in this section the adjusted revenue as the *RTB* revenue reduced by the penalty the publisher has to pay on the direct campaigns in case of under-delivery. In both experiments, we plot the adjusted revenue of the publisher when applying the estimated optimal strategy on the whole dataset, as a function of the number of batches of size 1000 processed by the algorithm.

6.1 Descriptive analysis of the dataset

The table below gives the main characteristics of the dataset, per placement and globally.

	Nb Imps	Avg <i>RTB</i> bid	Avg view rate	Avg click rate
\mathcal{P}_1	15.46 mlns	13.15 \$CPM	77.0 %	0.13 %
\mathcal{P}_2	5.56 mlns	15.36 \$CPM	77.9 %	0.23 %
\mathcal{P}_3	2.43 mlns	15.55 \$CPM	59.7 %	2.03 %
\mathcal{P}_4	0.14 mlns	25.37 \$CPM	42.7 %	0.00 %
Total	23.59 mlns	13.99 \$CPM	75.3 %	0.35 %

6.2 First experiment: optimization with a few campaigns

In this section, we define a direct campaign by the following sextuplet:

$$(N_{imp}, N_{view}, N_{click}, l_{imp}, l_{view}, l_{click})$$

, where N_* and l_* correspond to the goal and to a penalty parameter associated to the number of impressions, the number of impressions viewed and the number of impressions clicked. We assume that all campaigns target all the placements: this hypothesis makes the results simpler to interpret.

We assume that the penalty functions are rectified linear functions where l_* is the penalty per non-delivered impression. If v_{imp} , v_{view} and v_{click} are the volumes of impressions, of impressions viewed and of impressions clicked which have been delivered, the penalty function writes:

$$L(v_{imp}, v_{view}, v_{click}) = \sum_{i \in \{imp, view, click\}} l_i \times \max(N_i - v_i, 0)$$

This experiment runs the optimization algorithm with the following campaigns:

- 3 impressions campaigns:
 - $C_1 = (3.10^6, \dots, 5, \dots)$
 - $C_2 = (3.10^6, \dots, 10, \dots)$
 - $C_3 = (3.10^6, \dots, 20, \dots)$
- 3 viewability campaigns:
 - $C_4 = (., 2.10^6, \dots, 5, .)$
 - $C_5 = (., 2.10^6, \dots, 15, .)$
 - $C_6 = (., 2.10^6, \dots, 30, .)$
- 3 click campaigns:
 - $C_7 = (., ., 2.10^3, \dots, 200)$
 - $C_8 = (., ., 2.10^3, \dots, 500)$
 - $C_9 = (., ., 2.10^3, \dots, 1000)$

To run the experiment, we have used simple viewability and click predictors which predict the average viewability rate and click rate of the placement. These predictors are used before each auction to compute the scores of the campaigns. Using more advanced predictors based on more features would clearly improve the results.

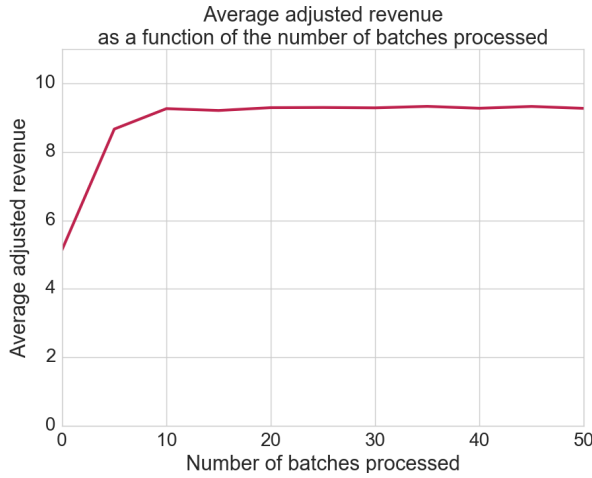


Figure 1: First experiment: average adjusted revenue as a function of the number of batches processed

Figure 1 plots the adjusted revenue estimated on the whole dataset as a function of the number of batches processed by the optimization algorithm. The initialization point corresponds to the case where no impression is sold in a direct campaign and it is very far from the optimum in terms of adjusted revenue. The algorithm converges after having processed a few dozens of batches.

The table below has been built by applying the optimal strategy estimated after having processed 100 batches of auctions. It gives, for each campaign, the number of impressions delivered on each placement, as a percentage of the goal of the campaign. It gives also the undelivered volume (i.e. the goal less the volume delivered) as a percentage of the goal of the campaign.

Campaign	P1	P2	P3	P4	Undelivered
C_1	12%	4%	1%	0%	83%
C_2	49%	15%	4%	0%	32%
C_3	73%	23%	5%	0%	0%
C_4	8%	3%	0%	0%	89%
C_5	70%	22%	2%	0%	6%
C_6	75%	23%	2%	0%	0%
C_7	0%	0%	19%	0%	81%
C_8	0%	0%	83%	0%	17%
C_9	0%	0%	100%	0%	0%

We see that the campaigns C_1 , C_2 , C_4 , C_5 , C_7 and C_8 are not fully delivered: for these campaigns, the penalty in case of under-delivery is low compared to the opportunity cost incurred by the publisher by not selling impressions on the *RTB* market. Also, the campaigns targeting clicks are mainly delivered on placement P_3 , which has a significantly higher click probability whereas the *RTB* bid is not significantly higher than on the other placements.

6.3 Second experiment: optimization with many campaigns

We note in this section $U(a, b)$ the uniform distribution on $[a, b]$. In this experiment, we run the optimization with K campaigns, and each campaign is generated the following way:

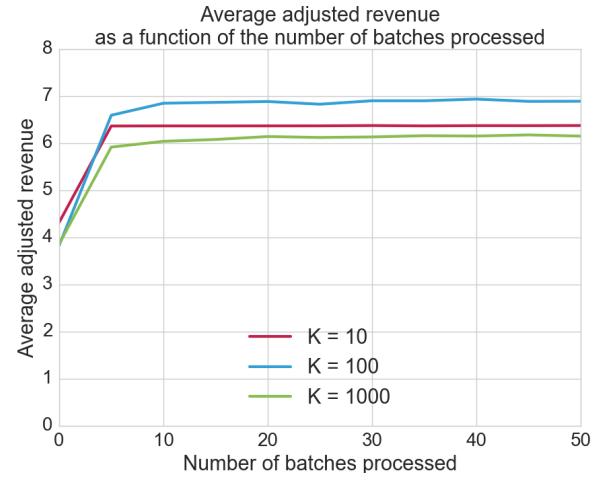


Figure 2: Second experiment: average adjusted revenue as a function of the number of batches processed

- It targets all impressions where $h(\text{imp_id}) \equiv 1 \pmod{M}$, where h is a hash function, imp_id is the id of the impression, and M is drawn from $U(10, 100)$. This methodology ensures that the targetings of a pair of campaigns has most of the time a non-empty intersection (which makes probably the optimization harder, as each impression must be allocated efficiently among all the campaigns targeting it)
- The impression goal N_{imp} is drawn from the distribution $U(0, N)$, where N is chosen such that the sum of the goals on all campaigns represent in average 40% of the total number of impressions available. Consequently, N decreases when K increases. Note that in this experiment, we do not define any goals on the number of impressions clicked or viewed
- The penalty parameter l_{imp} associated to the impressions goal is drawn from the distribution $U(0, 50)$

Figure 2 plots the adjusted revenue of the publisher on the whole dataset as a function of the number of batches processed by the optimization algorithm, for different values of the number of campaigns K .

We see on the figure that increasing the number of campaigns does not slow the convergence of the algorithm, and a few dozens of batches are enough to converge to the optimum. In practice, the optimum is attained in less than a few minutes with 1000 campaigns.

7 CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we study the problem of optimizing the revenue a web publisher gets through *RTB* and *Direct*. For each impression, the publisher bids in the *RTB* auction and chooses the direct campaign to deliver should it win the auction. We have been able to define an algorithm to determine an optimal strategy for the publisher in this setting.

The experiments performed on a *RTB* auctions dataset show that the algorithm scales with the number of campaigns, as increasing the number of campaigns does not slow the convergence. With 1000 campaigns, an optimal strategy can be estimated in a few minutes.

Also, applying the optimal strategy is particularly simple and the most computationally intensive part is to compute the score of each campaign, which is a weighted sum of the predictors $\theta_{n,k,i}$. In practice, the publisher has a few milliseconds to determine its bid. Should the number of campaigns be too high to be able to compute all the scores, the publisher should define a strategy to sample the campaigns considered for a given impression.

In practice, a web publisher should estimate the optimal strategy on a long period of time (ideally on several months) to make it as stable as possible. Applying it in the future relies on the hypothesis that the future *RTB* market is similar to the past auctions used in the estimation, in terms of volume and bid level. The estimation can also be adjusted to reflect an expected change in the *RTB* market.

When the *RTB* market characteristics differ too much from the dataset used in the estimation, the publisher should define a methodology to adjust the strategy. Such a methodology can be based for example on a controller which updates the strategy from time to time. This is a different topic which has been studied specifically in the literature (see [9], [8]).

The algorithm presented in the paper combined with a controller form a system which is being developed to be deployed on thousands of publishers worldwide.

Finally, the hypothesis that the set of campaigns is known when estimating the strategy is well suited to the theoretical analysis of the problem and to the derivation of an estimation algorithm. In practice, the publisher should make hypotheses on the campaigns which will be agreed in the future.

8 APPENDIX

8.1 Derivation of the optimality conditions

The lagrangian of the problem writes:

$$\begin{aligned} \mathcal{L}((a_n), (q_{n,k}), (v_{k,i})) = & \sum_{n=1}^N r_n(a_n) + \\ & \sum_{k=1}^K \left(\pi_k - L_k(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k}) \right) \\ & - \sum_{k=1}^K \sum_{i=1}^{p_k} \lambda_{k,i} (v_{k,i} - \sum_{n=1}^N F_n(a_n) q_{n,k} \theta_{n,k,i}) \\ & - \sum_{n=1}^N \mu_n \left(\sum_{k=1}^K q_{n,k} - 1 \right) + \sum_{n=1}^N \sum_{k=1}^K \eta_{n,k} q_{n,k} \end{aligned} \quad (7)$$

where $\lambda_{k,i}$, μ_n and $\eta_{n,k}$ are the lagrange multipliers corresponding to the constraints.

The Karush-Kuhn-Tucker optimality conditions are:

$$\begin{cases} \forall n, \frac{\partial \mathcal{L}}{\partial a_n} = & r'_n(a_n) + f_n(a_n) \sum_{k=1}^K q_{n,k} \left(\sum_{i=1}^{p_k} \lambda_{k,i} \theta_{n,k,i} \right) = 0 \\ \forall n, k, \frac{\partial \mathcal{L}}{\partial q_{n,k}} = & F_n(a_n) \sum_{i=1}^{p_k} \lambda_{k,i} \theta_{n,k,i} - \mu_n + \eta_{n,k} = 0 \\ \forall k, i, \frac{\partial \mathcal{L}}{\partial v_{k,i}} = & -\frac{\partial L_k}{\partial v_{k,i}}(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k}) - \lambda_{k,i} = 0 \end{cases} \quad (8)$$

8.1.1 Conditions on $q_{n,k}$. From the second equation we get for auction n :

$$\forall k, F_n(a_n) \sum_{i=1}^{p_k} \lambda_{k,i} \theta_{n,k,i} = \mu_n - \eta_{n,k}$$

Let's consider the set M_n of k -s maximizing $\sum_{i=1}^{p_k} \lambda_{k,i} \theta_{n,k,i}$:

$$M_n = \operatorname{argmax}_k \sum_{i=1}^{p_k} \lambda_{k,i} \theta_{n,k,i}$$

If $F_n(a_n) > 0$, we get:

$$M_n = \operatorname{argmax}_k (\mu_n - \eta_{n,k})$$

As $\eta_{n,k}$ is the Lagrange multiplier corresponding to the constraint $q_{n,k} \geq 0$, we have $\forall k, \min(\eta_{n,k}, q_{n,k}) = 0$. We know also that there must be at least one k such that $\eta_{n,k} = 0$ (otherwise all $q_{n,k}$ would be equal to 0 and they couldn't sum to 1).

We deduce that $\max_k (\mu_n - \eta_{n,k}) = \mu_n$, and finally: if $k \in M_n$, $\eta_{n,k} = 0$ (and $q_{n,k} > 0$), otherwise $\eta_{n,k} > 0$ (and $q_{n,k} = 0$).

We introduce the score of campaign C_k in auction n , defined by:

$$c_{n,k} = - \sum_{i=1}^{p_k} \theta_{n,k,i} \frac{\partial L_k}{\partial v_{k,i}}(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k})$$

We obtain the following optimality condition:

$$q_{n,k} > 0 \Rightarrow k \in \operatorname{argmax}_k c_{n,k}$$

8.1.2 Conditions on a_n . The first equation gives the optimality condition on a_n :

$$r'_n(a_n) + f_n(a_n) \max_k c_{n,k} = 0$$

8.2 Particular case where some penalty functions are rectified linear functions

We assume in this section that the penalty function for campaign C_k writes $L_k(v_{k,1}, \dots, v_{k,p_k}) = \sum_{i=1}^{p_k} l_i \max(g_{k,i} - v_{k,i}, 0)$. This kind of penalty function is widely used in practice but is not differentiable.

Let $\partial \mathcal{L}_{v_{k,i}}$ be the sub-gradient of the lagrangian \mathcal{L} with respect to $v_{k,i}$, which writes:

$$\partial \mathcal{L}_{v_{k,i}} = \begin{cases} \{-l_i - \lambda_{k,i}\} & \text{if } v_{k,i} < g_{k,i} \\ [-l_i - \lambda_{k,i}, -\lambda_{k,i}] & \text{if } v_{k,i} = g_{k,i} \\ \{-\lambda_{k,i}\} & \text{if } v_{k,i} > g_{k,i} \end{cases}$$

The Karush-Kuhn-Tucker optimality condition with respect to $v_{k,i}$ writes: $0 \in \partial \mathcal{L}_{v_{k,i}}$. This gives similar optimality conditions where the score of campaign C_k is written the following way:

$$c_{n,k} = \sum_{i=1}^{p_k} \theta_{n,k,i} \kappa_{k,i} \text{ with } \begin{cases} \kappa_{k,i} = l_i & \text{if } v_{k,i} < g_{k,i} \\ \kappa_{k,i} \in [0, l_i] & \text{if } v_{k,i} = g_{k,i} \\ \kappa_{k,i} = 0 & \text{if } v_{k,i} > g_{k,i} \end{cases}$$

8.3 Analysis of the optimality condition on a

In this section, we note F_b and F_c the cumulative probability functions of the highest bid B and the second highest bid C , and f_b and f_c their probability density functions. We assume that these functions are continuous on $[0, +\infty[$ and that $\forall a, f_b(a) > 0$.

The optimality condition on a states that the optimal value verifies $\frac{r'}{f_b}(a) = -c$, where $c \geq 0$ is the maximum score of the direct campaigns.

We show in this section that in a second-price auction mechanism, this optimality condition has at least one solution under a usual hypothesis on the distributions B and C . The solution is very simple in a first-price auction mechanism.

8.3.1 Second-price auction mechanism. We have:

$$r(a) = \mathbb{E}[\mathbf{1}_{a \leq B} \max(a, C)] = \mathbb{E}[\mathbf{1}_{a \leq C} C] + \mathbb{E}[\mathbf{1}_{C \leq a \leq B} a]$$

Note that $\mathbb{E}[\mathbf{1}_{C \leq a \leq B}] = P(a \leq B) - P(a \leq C) = F_c(a) - F_b(a)$, because $C \leq B$.

Then:

$$\begin{aligned} r'(a) &= F_c(a) - F_b(a) + a(f_c(a) - f_b(a)) - af_c(a) \\ &= -af_b(a) + (F_c(a) - F_b(a)) \end{aligned}$$

And:

$$\frac{r'}{f_b}(a) = -a + \frac{F_c(a) - F_b(a)}{f_b(a)}$$

It is worth noting that this function depends on the marginal distributions B and C only. It does not depend on the joint distribution (B, C) , which makes the analysis much simpler.

It is easy to see that $\frac{r'}{f_b}(0) = 0$ (as $F_b(0) = F_c(0) = 0$). To analyze the behaviour of this function when $a \rightarrow +\infty$, we write the function $\frac{r'}{f_b}$ using the hazard rates of B and C , $h_b = \frac{f_b}{1-F_b}$ and $h_c = \frac{f_c}{1-F_c}$:

$$\frac{r'}{f_b}(a) = -a + \frac{1}{h_b(a)} - \frac{1}{h_c(a)} \frac{f_c(a)}{f_b(a)} \leq -a + \frac{1}{h_b(a)}$$

We assume also that $\frac{1}{h_b(a)} = o(a)$. This hypothesis is quite usual and is verified by the normal and the log-normal distributions. Therefore, we have:

$$\lim_{a \rightarrow +\infty} \frac{r'}{f_b}(a) = -\infty$$

As $\frac{r'}{f_b}$ is a continuous function on $[0, +\infty[$, we conclude that under the hypothesis $\frac{1}{h_b(a)} = o(a)$, $\frac{r'}{f_b}$ is a surjective function to $[0, -\infty[$. This means that the optimality condition $\frac{r'}{f_b}(a) = -c$ has at least one solution for every $c \geq 0$. However, we have no guarantee on the unicity of the solution.

8.3.2 First-price auction mechanism. The first-price auction can be modeled with $C = B$, and in this case we have $\frac{r'}{f_b}(a) = -a$. The optimality condition has one unique solution $a = c$.

8.4 A regularization of the optimization problem

We introduce in this section a regularization of the optimization problem which makes the optimality conditions more stable. For each auction n , we add the term $\rho H(Q_n)$ to the objective function, where ρ is the regularization parameter which does not depend on n and $H(Q_n)$ is the entropy of the distribution Q_n . The case $\rho \rightarrow 0$ corresponds to the non regularized optimization problem, and in the case $\rho \rightarrow +\infty$, the optimal Q_n distribution is the uniform distribution.

The objective function of the problem becomes:

$$\begin{aligned} y_{regularized} \left((a_n)_{n=1}^N, (q_{n,1}, \dots, q_{n,K})_{n=1}^N, \rho \right) = \\ \sum_{n=1}^N r_n(a_n) + \sum_{k=1}^K \left(\pi_k - L_k(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k}) \right) \\ - \rho \sum_{n=1}^N \left(\sum_{k=1}^K q_{n,k} \ln(q_{n,k}) \right) \end{aligned} \quad (9)$$

The Karush-Kuhn-Tucker optimality conditions write:

$$\begin{cases} \forall n, \frac{\partial \mathcal{L}}{\partial a_n} = & r'_n(a_n) + f_n(a_n) \sum_{k=1}^K q_{n,k} c_{n,k} = 0 \\ \forall n, k, \frac{\partial \mathcal{L}}{\partial q_{n,k}} = & F_n(a_n) c_{n,k} - \mu_n - \rho(1 + \ln(q_{n,k})) = 0 \\ \forall i, k, \frac{\partial \mathcal{L}}{\partial v_{k,i}} = & -\frac{\partial L_k}{\partial v_{k,i}}(v_{k,1} - g_{k,1}, \dots, v_{k,p_k} - g_{k,p_k}) - \lambda_{k,i} = 0 \end{cases} \quad (10)$$

The second equation gives:

$$q_{n,k} = e^{-(1 + \frac{\mu_n}{\rho})} e^{c_{n,k} \frac{F_n(a_n)}{\rho}}$$

And the constraint $\sum_{k=1}^K q_{n,k} = 1$ gives:

$$q_{n,k} = \frac{\exp(c_{n,k} \frac{F_n(a_n)}{\rho})}{\sum_{k'=1}^K \exp(c_{n,k'} \frac{F_n(a_n)}{\rho})}$$

Finally, the optimality conditions of the problem are the following:

- The $q_{n,k}$ define a Boltzmann distribution with the weights $e^{c_{n,k}}$ and a temperature $\frac{\rho}{F_n(a_n)}$
- a_n verifies $r'_n(a_n) + f_n(a_n) \sum_{k=1}^K q_{n,k} c_{n,k} = 0$

REFERENCES

- [1] Santiago Balseiro, Jon Feldman, Vahab Mirrokni, and S. Muthukrishnan. 2011. Yield Optimization of Display Advertising with Ad Exchange. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC '11)*. ACM, New York, NY, USA, 27–28. <https://doi.org/10.1145/1993574.1993580>
- [2] Vijay Bharadwaj, Peiji Chen, Wenjing Ma, Chandrashekar Nagarajan, John Tomlin, Sergei Vassilvitskii, Erik Vee, and Jian Yang. 2012. SHALE: An Efficient Algorithm for Allocation of Guaranteed Display Advertising. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 1195–1203. <https://doi.org/10.1145/2339530.2339718>
- [3] Pedro Chahua, Nicolas Grislain, Gregoire Jauvion, and Jean-Michel Renders. 2017. Real-Time Optimization of Web Publisher RTB Revenues. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. ACM, New York, NY, USA, 1743–1751. <https://doi.org/10.1145/3097983.3098150>
- [4] B. Chen. 2016. PhD thesis, Financial Methods for Online Advertising. (2016).
- [5] B. Chen. 2016. Risk-Aware Dynamic Reserve Prices of Programmatic Guarantee in Display Advertising. (2016).
- [6] Bowei Chen, Shuai Yuan, and Jun Wang. 2014. A Dynamic Pricing Model for Unifying Programmatic Guarantee and Real-Time Bidding in Display Advertising. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising (ADKDD'14)*. ACM, New York, NY, USA, Article 1, 9 pages. <https://doi.org/10.1145/2648584.2648585>
- [7] A. Ghosh, P. McAfee, K. Papineni, and Vassilvitskii S. 2009. Bidding for Representative Allocations for Display Advertising. (2009).
- [8] N. Karlsson and J. Zhang. 2013. Applications of feedback control in online advertising. (2013).

- [9] Brendan Kitts, Michael Krishnan, Ishadutta Yadav, Yongbo Zeng, Garrett Badeau, Andrew Potter, Sergey Tolkachov, Ethan Thornburg, and Satyanarayana Reddy Janga. 2017. Ad Serving with Multiple KPIs. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. ACM, New York, NY, USA, 1853–1861. <https://doi.org/10.1145/3097983.3098085>
- [10] Kuang-Chih Lee, Ali Jalali, and Ali Dasdan. 2013. Real Time Bid Optimization with Smooth Budget Delivery in Online Advertising. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising (ADKDD '13)*. ACM, New York, NY, USA, Article 1, 9 pages. <https://doi.org/10.1145/2501040.2501979>
- [11] Pavankumar Murali, Ying Li, Pietro Mazzoleni, and Roman Vaculin. 2015. Optimal Budget Allocation Strategies for Real Time Bidding in Display Advertising. In *Proceedings of the 2015 Winter Simulation Conference (WSC '15)*. IEEE Press, Piscataway, NJ, USA, 3146–3147. <http://dl.acm.org/citation.cfm?id=2888619.2889011>
- [12] Michael Ostrovsky and Michael Schwarz. 2011. Reserve Prices in Internet Advertising Auctions: A Field Experiment. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC '11)*. ACM, New York, NY, USA, 59–60. <https://doi.org/10.1145/1993574.1993585>
- [13] G. Roels and K. Fridgeirsdottir. 2009. Dynamic Revenue Management for Online Display Advertising. (2009).
- [14] Jun Wang and Bowei Chen. 2012. Selling Futures Online Advertising Slots via Option Contracts. In *Proceedings of the 21st International Conference on World Wide Web (WWW '12 Companion)*. ACM, New York, NY, USA, 627–628. <https://doi.org/10.1145/2187980.2188160>
- [15] Shuai Yuan, Jun Wang, Bowei Chen, Peter Mason, and Sam Seljan. 2014. An Empirical Study of Reserve Price Optimisation in Real-time Bidding. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 1897–1906. <https://doi.org/10.1145/2623330.2623357>