# Bid Shading by Auction Clearing Price Prediction and Fast Surplus Maximization Search

Shengjun Pan, Brendan Kitts, Tian Zhou, Hao He, Bharatbhushan Shetty, Aaron Flores, Djordje Gligorijevic, Junwei Pan, Tinyu Mao, San Gultekin and Jianlong Zhang

## ABSTRACT

This paper describes a new Bid Shading algorithm, called "Win Rate", that is currently used in a large online advertising company. The method uses a modified logistic regression to predict the profit from each possible shaded bid price. The function form allows fast maximization at run-time, a key requirement for Real-Time Bidding systems. We report production results from this method along with several other algorithms. We find that bid shading, in general, can deliver significant value to advertisers, reducing price per impression to about 55% of the unshaded cost. Further, the particular approach described in this paper captures 7% more profit for advertisers, than do benchmark methods of just bidding the most probable winning price. We also report 4.3% higher surplus than an industry Sell-Side Platform shading service. We attribute the gains above as being mainly due to the explicit maximization of the surplus function, and note that other algorithms can take advantage of this same approach.

## CCS CONCEPTS

• **Applied computing → Online auctions**; • **Information systems → Display advertising**; • **Computing methodologies → Machine learning algorithms**.

## KEYWORDS

online bidding, shading, auction, advertising, bid, optimization

**ACM Reference Format:**
Shengjun Pan, Brendan Kitts, Tian Zhou, Hao He, Bharatbhushan Shetty, Aaron Flores, Djordje Gligorijevic, Junwei Pan, Tinyu Mao, San Gultekin and Jianlong Zhang. 2018. Bid Shading by Auction Clearing Price Prediction and Fast Surplus Maximization Search. In *AdKDD 2020 Workshop held at KDD 2020: SIG Conference on Knowledge Discovery and Data Mining, August 22–27, 2020, San Diego, CA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

Online Advertising auctions have been dominated by Second Priced Auctions since their early implementations in the 1990s. Google famously used Second Price Auctions for its Adwords and Adsense auctions, and, in 2017, generated 90% of its revenue from Second Price Auctions [17]. However, there was a dramatic shift in online advertising between 2018 and 2019. As of 2020, almost all major display ad auctions have switched from Second to First Price Auctions [18, 19]. Several factors conspired to drive the industry towards the adoption of FPA, including the widespread growth of header bidding with its incompatibility with SPAs [23], increased demand for transparency and accountability [12, 16, 28, 31], and yield concerns [5],[27],[6].

Unfortunately for advertisers, First Price Auctions leave private value bidders susceptible to over-paying. For instance, if the bidder's private value of an impression was \$10.00, and the winner knew the second placed bidder's price was just \$1.00, they could bid just \$1.01 and effectively collect a \$8.99 profit. If they instead bid their private value, they would be charged the entirety of the \$10.00 and they would have \$0 profit!

The practice of strategically decreasing bid price below the buyer's private value is known as *bid shading*. Bid shading has been observed in a variety of real world auctions including FCC Spectrum [11], US Oil Deposits [10], Cattle auctions [13], US Treasury auctions [22] and others. Despite its widespread use, there has been little work done on methods to systematically exploit shading, particularly when data is available to make it possible to predict auction clearing prices.

## 2 THE BID SHADING PROBLEM

Given impression $i$, and a valuation for the impression $V_i$ which represents how much the advertiser expects to capture from the impression, how much should the advertiser discount their valuation? Assuming that the valuation $V_i$ is an accurate representation of the dollar value that the advertiser expects to obtain, and the bid $b_i = g_i V_i$ is also in real dollars, the advertiser's financial gain or surplus is equal to:

$$\text{bid surplus} = \sum_{i=1}^{N} (V_i - g_i V_i)\mathbf{I}(g_i V_i), \tag{1}$$

where $g_i$ is the shading factor to apply to the bidder's private value $V_i$, $\hat{b}_i$ is the auction price needed to win, and $\mathbf{I}(b_i) = 1$ , if $b_i \geq \hat{b}_i$ , and 0 otherwise. The task is to find a shading factor $g_i \in (0..1)$ that maximizes the surplus to the advertiser.

## 3 PREVIOUS WORK

### 3.1 Bid Shading Theory

Bid shading is a common tactic in repeated First Price Auctions. [35] finds robust evidence of shading in Austrian livestock auctions and [13] reported shading in a Texas cattle market. [22] find the practice in auctions for US Treasury notes.

Auctions generally need to be repeated and predictable for bid shading to be practically feasible, but under these conditions, it often occurs organically. Pownall and Wolk (2013) showed that bid

---

*AdKDD 2020 Workshop, 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2020), August 22–27, 2020, San Diego, CA*

ACM ISBN 978-1-4503-XXXX-X/18/06…\$15.00
https://doi.org/10.1145/1122445.1122456

shading for repeated internet auction prices increased over time; by about 26% after 10 iterations [26]. When there are enough repeated games bidders can even develop collusive shading strategies where bidders actively coordinate to have low bids [25] [21].

Although behavior varies from auction to auction, several studies have shown that the magnitude of shading tends to increase with the average price on the auction [11] [9] [22]. This is likely to occur because of the more substantial losses involved on higher priced auctions, if shading isn't sufficient. This result suggests that using a measure of the expense of the auction is valuable when trying to estimate the shading factor - a finding we revisit later in Section 6.

In situations where the supply is plentiful, and demand limited, buyers can shade deeper. In looking at this phenomemon in the US Treasury Market, Hortacsu et. al. (2017) find that large institutional buyers on average shade more aggressively than small indirect buyers [22] . This seems to be because these large buyers effectively control a large percentage of bidders, and so it is almost like they are able to "coordinate" the buying of multiple buyers. They can therefore drive the bid prices for a large percentage of bidders down, whilst still meeting their goals.

## 3.2 Previous Algorithms

In 2018 and 2019, Rubicon [4], [27], AppNexus [5] and Google [30] [32] [19] [20] all released Sell-Side bid shading services. Leading up to this, there had been reports of dramatically lower ROI from the new First Price Auctions [23],[6]. Never-the-less, this is a surprising move as Sell-Side Platforms are potentially decreasing their yield, and they clearly have a different incentive from buyers. The sell-side algorithms seem to reflect this incentive difference. The descriptions of these services suggest that they try to keep bid prices high enough to maintain a set win-rate, but preventing the bid price from becoming too extreme; which might risk an advertiser to halt their bidding due to poor Return on Investment. Rubicon released data suggesting that their service decreases First Price CPMs by a modest 5% over 4 months [27]. AppNexus reported that prices under their service were 25% lower over 100 days [5]. We tried one of the services, and recorded the shading distribution in Figure 3. Most of the bid shades were about 90%, which is conservative for our problem. Further analysis on Sell-Side "Bid Shaders" are in Section 7.

On the Demand Side, a variety of algorithms have been explored, although generally not exactly for bid shading applications. [34] developed a "censored winning bid probability estimator". They observed that when a bidder submitted a bid and lost, the information gained is that the winning price is somewhere above the submitted price; and when a bidder submits a bid and wins, the minimum bid to win is at a price somewhere below their submitted bid. Using these two cases, the authors developed a Maximum Likelihood procedure to estimate the probability of the winning bid being any of the bid prices. This created a distribution of the probable winning bids, with the most likely winning bid price being used for shading. [33] extend their work to using a neural network to estimate the parameters of the win probability distribution.

[8] used Linear Regression to predict the minimum winning bid price $\hat{b}$ using features in the request. The predicted clearing price was then used as the shaded bid price.

The approaches described above [34],[33],[8] all focus on predicting the probable winning bid price. However, the surplus maximum is very different from the minimum bid to win. An accurate (unbiased, symmetric noise) win probability estimator will be below the winning bid price about 50% of the time - this means that 50% of the surplus won't be captured *by design.* If the change in new impressions captured at a higher bid price, over-weights the marginal decrease in profitability per impression, the optimum for surplus can be higher than the most probable bid.

Unpublished work by [24] is one of the few that we know of to attempt to explicitly maximize the surplus function. These authors estimate shading factors for a set of fixed segments based on three bid samples taken in real-time to estimate the local surplus landscape. However the approach has many drawbacks: The segments have to be predetermined and finding a suitable segment definition requires substantial analysis. The information across segments is not shared, which is a problem for segments that do not have enough traffic. Further, the set of possible segments quickly explode as the number of variables used to define them increases. The approach taken in this paper uses a model to estimate the surplus function, and so a very large number of features can be used, and model induction is also automated, easy to maintain, and improve.

In order to compare the method we used to prior work, we have included an implementation of the Linear Regression algorithm from [8], the Distribution Estimator algorithm from [34], and the Segment-based Surplus maximizer [24] in the benchmarks which we use to analyze algorithm performance in Section 7.

# 4 CANONICAL ALGORITHM

Given a bid request for first-price auction, let $x_1, x_2, \ldots, x_k$ be the set of publisher and user attributes that we will use to find the best bid price $b^* < V$ to submit for the first-price auction. Let $\hat{b}$ be the highest bid price from other competing bidders, which value is unknown. Note that $\hat{b}$ depends on both attributes $x_i$s which represent the item that is being auctioned, and external competing bidder behavior. $\hat{b}$ follows an unknown distribution $\mathcal{D}_{\hat{b}|x_1,x_2,\ldots,x_k}$ with cumulative probability distribution $\mathrm{cdf}_{\hat{b}|x_1,x_2,\ldots,x_k}$. When the context is clear, we use $\mathcal{D}_{\hat{b}}$ and $\mathrm{cdf}_{\hat{b}}$ for simplicity.

If the distribution $\mathcal{D}_{\hat{b}}$ is known, we can calculate the optimal bid price $b^*$ directly as follows. Let $\mathbb{I}(b > \hat{b})$ be 1 if $b > \hat{b}$ and 0 otherwise, which indicates if the submitted price $b$ wins the auction. Then the surplus when the submitted price is $b$ is

$$surplus = (V - b)\mathbb{I}(b > \hat{b}) = \begin{cases} V - b, & \text{if } b > \hat{b}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The optimal bid price can be calculated as the price that maximizes the expected surplus

$$\begin{aligned} b^* &= \arg\max_{b>0} \mathbb{E}[surplus] \\ &= \arg\max_{b>0} \mathbb{E}\left[(V - b)\mathbb{I}(b > \hat{b})\right] \\ &= \arg\max_{b>0} (V - b)\,\mathrm{cdf}_{\hat{b}}(b). \end{aligned} \tag{3}$$

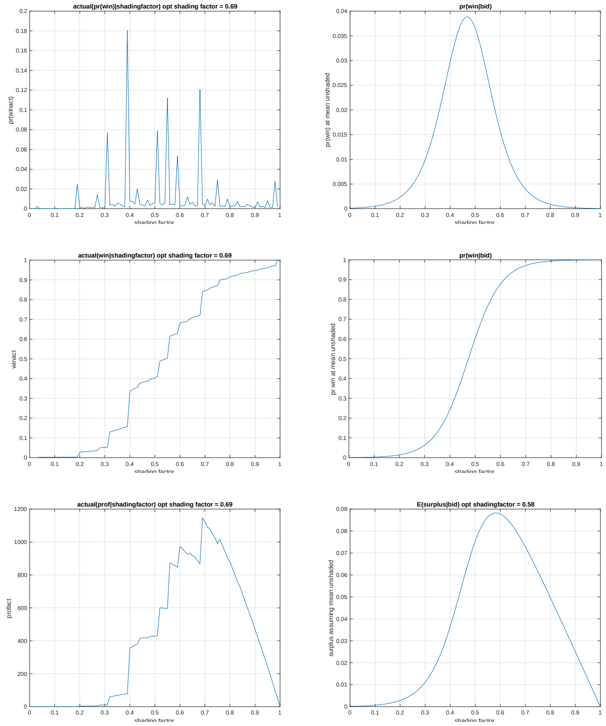

**Figure 1: Top: Actual PDF for $\hat{b}$ (left) versus estimate (right); middle: CDF actual versus estimate; bottom: Surplus distribution actual versus estimate.**

For simple forms of $F$, the optimization problem (3) can be solved analytically. Suppose $\hat{b}$ distributes uniformly over the interval $[B_0, B_1]$, where $0 \le B_0 < B_1$. This produces a $\mathrm{cdf}_{\hat{b}}(b)$ that is piece-wize linear, with a flat region of 0.0 from $0..B_0$, a constant slope from $B_0..B_1$, and another flat region of 1.0 above $B_1$. The bid price $b^*$ that maximizes surplus can be calculated as below

$$\begin{aligned}\mathbb{E}[surplus] &= (V-b)\,\mathrm{cdf}_{\hat{b}}(b)\\ &= \begin{cases} 0, & \text{if } b < B_0,\\ (V-b)(b-B_0)/(B_1-B_0), & \text{if } B_0 \le b \le B_1,\\ V-b, & \text{if } b > B_1.\end{cases}\end{aligned}$$

$$\max \mathbb{E}[surplus] = \begin{cases} \frac{(V-B_0)^2}{4(B_1-B_0)} \text{ at } b^* = \frac{V-B_0}{2}, & \text{if } V \le 2B_1 - B_0,\\ V - B_1 \text{ at } b^* = B_1, & \text{if } V > 2B_1 - B_0.\end{cases}$$

However, in practice, we rarely see such simple form of distributions. Figure 1 shows the empirical PDF of $\hat{b}$ for an example ad, including the derived surplus distribution. Our approach breaks into two steps:

1. Estimate the distribution $\mathcal{D}_{\hat{b}|x_1,x_2,\dots,x_k}$;
2. Solve the maximization problem (3).

## 4.1 Distribution Estimation

Given publisher and user attributions $x_1, \dots, x_k$ and bid price $b$, we use training a classification model with historical data:

$$\Pr(\mathrm{win}) = \mathrm{cdf}_{\hat{b}}(b) = F\left(w_0 + \sum_{i=1}^{k} w_i x_i + \beta\, g(b)\right), \tag{4}$$

where $F$ is a fitting function that outputs a value between 0 and 1, which must be monotonically increasing in $b$ (higher bid price leads to higher winning rate), and $g(b)$ is a bid transformation function such that $F \longrightarrow 0$ as $b \longrightarrow 0$, that is, as bid price goes to 0, the winning probability also goes to 0, and the weights to be learned are $w_0, w_1, \dots, w_k$ and $\beta$. There are different choices of functions $F()$ and $g(b)$. For $g(b)$, in this paper, we use the log of bid price $g(b) \overset{\mathrm{def}}{=} \log(b) \longrightarrow -\infty$, as $b \longrightarrow 0$. The choice of $F$ needs to satisfy the condition that $F(x) \longrightarrow 0$ as $x \longrightarrow -\infty$. We use the logistic function for this purpose [29],[14].

$$\Pr(\mathrm{win}) = \mathrm{logistic} = \left(1 + e^{-(w_0 + \sum_{i=1}^{k} w_i x_i + \beta \log b)}\right)^{-1}. \tag{5}$$

The goodness of fit is usually expressed a log likelihood function, and the parameters can then be trained by gradient descent [14]. However this formulation simply focuses on the quality of fit. In this application, we don't care as much about specific cases being fit, but rather, whether the surplus function is being predicted accurately. In other words, we want to minimize error for the surplus prediction in Figure 1. To do this, we change error to *Squared Surplus Error* where $y(b)$ is the actual win/loss and $y^*(b)$ is the prediction:

$$\mathrm{E} = [(V-b)(y(b) - y^*(b))]^2 \tag{6}$$

Differentiating error with respect to each parameter results in the function below:

$$\mathrm{E'}(w_i) = \frac{2x_i(V-b)a((V-b)(y(b)-y^*(b)))}{(a+1)^2} \tag{7}$$

where

$$a = e^{-(w_0 + \sum_{i=1}^{k} w_i x_i + \beta \log b)} \tag{8}$$

It is now possible to use gradient descent to numerically fit the parameters to minimize surplus error. $w_i = w_i - \epsilon\, \mathrm{E'}(w_i)$. This addresses some problems with organic win-loss data, such as the tendency for the fit to be dominated by economically less valuable auctions and landscape regions. Now model resources are orientated towards producing a better surplus estimate. We call this "Profitable Logistic Regression" or "Progistic Regression".

## 4.2 Surplus Maximization

The optimal bid price $b^*$ can now be found by solving the optimization (3):

$$\begin{aligned} b^* &= \underset{b>0}{\arg\max}(V-b)\,\mathrm{logistic}\left(w_0 + \sum_{i=1}^{k} w_i x_i + \beta \log b\right)\\ &= \underset{b>0}{\arg\max}(V-b)\left(1 + e^{-w_0 - \sum_{i=1}^{k} w_i x_i - \beta \log b}\right)^{-1}\\ &= \underset{b>0}{\arg\max} \frac{V-b}{1 + e^{-\alpha} b^{-\beta}}, \end{aligned} \tag{9}$$

where $\alpha = w_0 + \sum_{i=1}^{k} w_i x_i$.

We show below that, for $b > 0$, there is a single optimum bid price $b^*$ which can be bounded from above and below. These bounds make it possible to implement a fast bisection search.

Theorem 1. *For any* $\beta > 0$,

$$f(b) = \frac{V - b}{1 + e^{-\alpha} b^{-\beta}}$$

*is maximized at some unique* $b^*$ *such that*

$$\frac{\beta}{\beta + 1 + e^{\alpha} V^{\beta}} V \le b^* < \frac{\beta}{\beta + 1} V.$$

Proof. Taking the derivative, we have

$$f'(b) = \frac{\beta V - (\beta + 1) b - e^{\alpha} b^{\beta + 1}}{(1 + e^{-\alpha} b^{-\beta})^2 e^{\alpha} b^{\beta + 1}}.$$

For any $b \in (0, V]$, the above numerator can be bounded as

$$\beta V - (\beta + 1) b - e^{\alpha} b^{\beta+1} \ge \beta V - (\beta + 1) b - e^{\alpha} V^{\beta} b,$$
$$\beta V - (\beta + 1) b - e^{\alpha} b^{\beta+1} < \beta V - (\beta + 1) b,$$

and hence

$$\frac{\beta V - \left(\beta + 1 + e^{\alpha} V^{\beta}\right) b}{(1 + e^{-\alpha} b^{-\beta})^2 e^{\alpha} b^{\beta+1}} \le f'(b) < \frac{\beta V - (\beta + 1) b}{(1 + e^{-\alpha} b^{-\beta})^2 e^{\alpha} b^{\beta+1}}.$$

Then it's easy to verify that

$$f'\left(\frac{\beta}{\beta + 1 + e^{\alpha} V^{\beta}} V\right) \le 0 < f'\left(\frac{\beta}{\beta + 1} V\right).$$

Note that $f'(b)$ is a monotonically decreasing function. It follows that there is a unique $b^* \in \left[\frac{\beta}{\beta+1+e^{\alpha}V^{\beta}} V, \frac{\beta}{\beta+1} V\right)$ such that $f'(b^*) = 0$, that is, $f(b)$ is maximized at $b = b^*$. □

Theorem 1 allows us to implement a fast bisection search 4.1 for the optimal bid price. Starting with the minimum and maximum bounds on the surplus optimum, $b_{\min} = \frac{\beta}{\beta+1+e^{\alpha}V^{\beta}} V$ and $b_{\max} = \frac{\beta}{\beta+1} V$, (per Theorem 1), we know that the lower bound for optimum has positive derivative, and the high bound has negative. Bisection can divide the range and find the zero point for the derivative in O(log $N$) time; this is extremely desirable since the maximization search must run in real-time in the ad-server. We found in practice that we could use the gradient information to speed up the search further. Rather than cutting the range in half each time ($r = 0.5$; step 8), after testing the gradient of the minimum and maximum bid points, we use our knowledge that the surplus function is convex and so derivatives shorten close to the optimum. We calculate the ratio between the surplus derivative at min and max bid locations, and then use that estimate for the relative distance to the optimum in bid space. Step 8 and 9 of the pseudo-code show this modification to $r$. We ran a test with all data for May 21, 2020. The standard bisection search required on average 8.52 steps per bid request before terminating. Using the gradient estimate, the time decreased to 6.89 per request.

**Algorithm 4.1** Bisection Algorithm Surplus Maximization

```
Require:
 1: • Model weights: $w_0, w_1, \ldots, w_k, \beta$;
    • Feature values $x_1, x_2, \ldots, x_k$;
    • $V$: expected value of the current ad opportunity
    • $\epsilon > 0$: minimum valid interval length
    • $N$: maximum number of search steps
Ensure: $\beta > 0, V > 0$

 2: $\alpha \leftarrow w_0 + \sum_{i=1}^{k} w_i x_i$.
 3: $b_{\min} \leftarrow \frac{\beta}{\beta+1+e^{\alpha}V^{\beta}} V$
 4: $b_{\max} \leftarrow \frac{\beta}{\beta+1} V$
 5: for $i = 1, 2, \ldots, N$ do
 6:     $\mathrm{fp}_{\min} \leftarrow \beta V - (\beta + 1) b_{\min} - e^{\alpha} b_{\min}^{\beta+1}$
 7:     $\mathrm{fp}_{\max} \leftarrow \beta V - (\beta + 1) b_{\max} - e^{\alpha} b_{\max}^{\beta+1}$
 8:     $r \leftarrow -\mathrm{fp}_{\min} / (\mathrm{fp}_{\max} - \mathrm{fp}_{\min})$
 9:     $b \leftarrow (1 - r) b_{\min} + r\, b_{\max}$
10:     $\mathrm{fpb} \leftarrow \beta V - (\beta + 1) b - e^{\alpha} b^{\beta+1}$
11:     if fpb $< 0$ then
12:         $b_{\min} \leftarrow b$
13:     else
14:         $b_{\max} \leftarrow b$
15:     end if
16:     if $b_{\max} - b_{\min} < \epsilon$ then
17:         break
18:     end if
19: end for
    return $b$
```

## 5 IMPLEMENTATION

The features used for predicting win probability comprise 12 variables extracted from the HTTP of an incoming bid request, along with log(bid price) and log(bid price before shading). The HTTP attributes include the requesting page" (eg. "cnn.com/finance"), "request publisher" (eg. "cnn"), "device type" - desktop, mobile, tablet; "hour of day"; "day of week"; "country"; "user segment"; and other variables. All of the HTTP features are encoded to be 0-1 variables.

7 days of data in the past were used to train 1 day of data in the future. The training data size was 1.2 Billion rows of data with approximately 56,000 features. For the curve fit, we used the LogisticRegression method that is part of the PySpark pyspark.ml.classification library [3]. Training occured nightly and takes approximately 8 to 10 hours.

At run-time, the Bid Shader needs to respond to 5.5 million requests per second peak load, within 100 miliseconds for all systems[1]. In order to meet these speed constraints, bid shading has to minimize the number of computations that it performs. In terms of memory, by using a single global model, memory consumption is kept to just 56, 000 floating point numbers. In terms of time, shading optimization averages just 19 operations per request.

---

[1] 5.5 million requests per second peak, 800,000 per second average; 1 million responses per second at peak load, with 90,000 per second average. Given 750 bid servers, that means each server has to handle 5,000 requests per second. Overall, less than 10 miliseconds are budgeted for bid shading.

## 6 SHADING INSIGHTS

The log of bid price before shading and log of bid price are both highly predictive[2] ($\beta$ = -0.39 and 0.565; McFadden $R^2$=0.24 and 0.20 respectively [15],[1]). The high predictiveness of "bid price before shading" - and yet negative sign when included with bid price - is consistent with previous observations that bid shading tends to be deeper in auctions with higher valuations [11] [9] [22].

The top 0-1 feature in terms of impact on win probability is "is new user" ($\beta = 0.831$; $Pr$=0.52), which is associated with an increase in chance of winning the auction (since bid prices are lower). Hour of day 6am ($\beta = -0.267$; $Pr$=0.01) is associated with a drop in the probability of winning, likely due to the reduction in supply [7]. Country US ($\beta = -0.110$; $Pr$=0.84) decreases the chance of winning; and the largest 768x1024 ads also are less likely to be won ($\beta = -0.267$; $Pr$=0.01).

The predictability of time, user, and other features, for estimating auction clearing prices, suggest that shading should be effective, as noted in work on the preconditions for shading in Section 3 [26].

## 7 COMPARISON TO BENCHMARKS

We ran several of the algorithms in Section 3 as benchmarks. These included: (1) Sell-Side Shading Service (SSP) [4], [27], [5], [19], [20], (2) Non-linear segment-based (NL) [24], Distribution estimator with Normal (NRML), Exponential (EXP) Distributions [34], Linear Regression [8] and Unshaded (Uns). Win-Rate is labeled "WR" in the tables to follow.

The prior work benchmarks aren't ideal - the win distribution approaches [34] don't explicitly maximize surplus and so we expect them to not perform as well. The SSP services seem to be geared towards maintaining win-rate. Never-the-less we have included them, both to compare to prior work, but also to quantify the gain that surplus maximization approaches can deliver in practice.

Unlike the other benchmarks, the Segment-based algorithm does maximize surplus [24]. Under a favorable selection of segments, the Segment-Based Method might even be tuned to perform as well or better than the current method (despite the scaling problem with using more features). Our purpose in showing these benchmarks isn't to claim that this particular algorithm is "the best", but rather to show that Surplus Maximizers have an advantage, to quantify the gain, and to note that WR, which is fully automated, uses all available features to estimate the surplus landscape, and has excellent memory and speed properties, performs comparable to other reported approaches.

The experiments below (except ones with the SSP service) were run on auctions for which the minimum bid prices to win were known. Using this data it was possible to calculate surplus performance as a percentage of the optimal surplus. The algorithms were tested on a day of saved auction data from May 21, 2020. Training took 6 hours and May 22 was used for evaluation. 100% of the bid requests are scored by each algorithm, so all algorithms operate on the same set of records. The results are shown in Table 1.

The Distribution estimator methods (Nrm, Exp) estimate the minimum bid to win and so are not expected to do well in maximizing surplus. As a group they were about 7% below WR. The Nonlinear Segment method generated the second highest surplus besides WR. This makes sense given that it is a legitimate surplus maximizer. WR generates the highest surplus (50.6%). In sum, the Surplus Maximizers produced the most surplus, which was expected.

**Table 1: Benchmark Algorithms**

| Metric | WR | NL | Nrm | LR | Exp | Uns |
|---|---|---|---|---|---|---|
| %opt surp | 50.6% | 49.0% | 48.0% | 47.3% | 46.0% | 0% |
| %opt spen | 41.7% | 56.0% | 42.7% | 39.8% | 31.1% | 176% |
| %opt imps | 56.6% | 49.1% | 53.1% | 50.3% | 42.6% | 100% |
| shad fact | 0.6 | 0.55 | 0.62 | 0.61 | 0.42 | 1.00 |
| CPM | 1.06 | 1.64 | 1.16 | 1.14 | 1.05 | 2.52 |

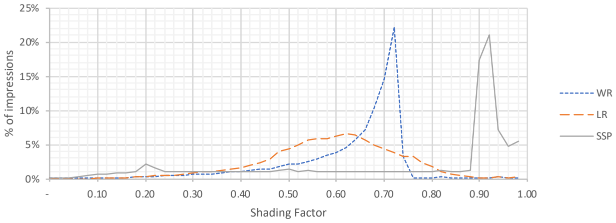

**Figure 2: Shading factor distributions for three algorithms. SSP has more shallow shading factors.**

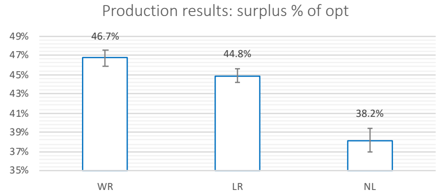

**Figure 3: Production Surplus Performance by Algorithm.**

We also compared an anonymous SSP Shading Service. We had to separate this analysis due to a service issue. When using the SSP Shading Service for real-time bidding, the service disabled the minimum bid to win functionality (!) As a result, we were unable to do an optimality analysis.

Overall, the SSP Shader delivered about 15% more impressions than WR - as noted SSPs have an incentive to try to monetize more traffic. However it delivered about 4.3% lower surplus. The bidding distribution from the Sell-Side Service is shown in Figure 2. Where-as the SSP's shading distribution is right-skewed, with most shading at 90% and above, the WR distribution - which generates more surplus - is left-skewed, with most shades below 72%. It seems likely that the SSP algorithm is geared towards generating high sales, but not necessarily high advertiser surplus.

## 8 PRODUCTION RESULTS

After rolling out the WR algorithm, we were able to monitor its performance by maintaining a percentage of traffic that was randomly allocated to each algorithm. The analysis spans from March 18 to May 6 2020 and is shown in Table 2. WR captured 46.7% of the maximum possible surplus, where-as Non-linear captured 38%. Bid prices on WR were about 45% lower than their unshaded prices.

---

[2] In the following, the regression coefficient is labeled $\beta$ and $Pr$ is the percentage of observations where the 0-1 variable is 1

**Table 2: Production Results**

| Metric | WR | LR | NL | Uns |
|---|---|---|---|---|
| % opt surp | 46.7% | 44.8% | 38.2% | 0.0% |
| % opt spend | 79.1% | 72.6% | 89.9% | 410% |
| % opt imps | 60.3% | 51.4% | 56.0% | 100% |
| shad fact | 0.55 | 0.53 | 0.59 | 1.00 |
| bid price | 1.13 | 1.02 | 1.21 | 2.05 |
| $\sigma$ % opt surplus | 2.9% | 2.3% | 4.4% | 0.0% |
| $\sigma$ % opt spend | 7.0% | 3.7% | 24.5% | 129% |
| $\sigma$ % opt imps | 3.7% | 1.9% | 10.5% | 0.0% |
| $\sigma$ shad fact | 0.022 | 0.030 | 0.076 | 0.000 |
| $\sigma$ bid price | 0.166 | 0.108 | 0.287 | 0.899 |
| days | 49 | 42 | 50 | 43 |

# 9 CONCLUSION

There is evidence that First Price Auctions have created problems for advertisers. Average traffic prices are higher, with estimates ranging between 5% and 50% [27], [6], [5], [23]. [6] also reported that after their SSP switched to First Price, 10% of advertisers actually discontinued bidding. Our experiments confirm these findings; without a shading solution, CPM would approximately double.

DSPs are required to compute the private value of impressions based on advertiser parameters, and they also execute a large number of trades, and so can build up an ability to predict auction prices. This makes it possible to implement rational shading similar to other industries [25] [21], [22]. Advertiser bids follow the value of traffic, and this follows daily, hourly, and site patterns. As a result, auction prices will always have structure that can be used by some advertisers with other advertisers have less flexibility.

The Surplus Maximization approach of this paper delivered about 7% higher surplus than naive methods just designed to submit the probable clearing price. Publicly available data shows medium sized DSPs managing between $260 to $1 Billion in advertiser spend [2]. The Shading gains reported in this paper therefore represent $18 to $100 million in additional yield that is provided to advertisers. Shading has an enormous impact on advertiser profitability. Now that the online Ad Industry has increasingly shifted to First Price Auctions, it seems likely that the new advertising technology arms race will be in the domain of Bid Shading.

# REFERENCES

[1] 2011. *What are Pseudo R Squares?* https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/

[2] 2017. *Worldwide Digital Advertising Software Market Shares 2017.* https://www.criteo.com/wp-content/uploads/2018/09/US44240218e_Criteo.pdf

[3] 2020. *Source code for pyspark.ml.classification.* https://spark.apache.org/docs/2.1.1/api/python/_modules/pyspark/ml/classification.html

[4] AdExchanger. 2017. Rubicon Joins First-Price Auction Club; Diageo Is Latest Brand To Demand More Transparency. *AdWeek.* https://adexchanger.com/ad-exchange-news/tuesday-12122017/

[5] AppNexus. 2018. Demystifying Auction Dynamics for Digital Buyers and Sellers. *AppNexus Website.* https://www.appnexus.com/sites/default/files/whitepapers/49344-CM-Auction-Type-Whitepaper-V9.pdf

[6] author removed. 2019. Bidder Behavior after Shifting from Second to First Price Auctions in Online Advertising. *unpublished.* http://www.appliedaisystems.com/papers/FPA_Effects33.pdf

[7] author removed. 2019. Lookahead Algorithms for Online Bidding. In *in review.*

[8] authors removed. [n.d.]. Bid Shading in The Brave New World of First-Price Auctions. In *in review.*

[9] Pierpaolo Battigalli and Marciano Siniscalchi. 2003. Rationalizable bidding in first-price auctions. *Games and Economic Behavior* 45, 1 (2003), 38–72.

[10] R.V. Clapp Capen, E.C. and W.M. Campbell. [n.d.]. Auctions and Bidding. *Journal of Petroleum Technology* 23, 6 ([n. d.]), 641–643.

[11] Bhaskar Chakravorti, William W Sharkey, Yossef Spiegel, and Simon Wilkie. 1995. Auctioning the airwaves: the contest for broadband PCS spectrum. *Journal of Economics & Management Strategy* 4, 2 (1995), 345–373.

[12] V Chari and Robert Weber. 1992. How the US Treasury should auction its debt. *Federal Reserve Bank of Minneapolis Quarterly Review* 16, 4 (1992). http://kylewoodward.com/blog-data/pdfs/references/chari+weber-quarterly-review-1992A.pdf

[13] John M Crespi and Richard J Sexton. 2005. A Multinomial logit framework to estimate bid shading in procurement auctions: Application to cattle sales in the Texas Panhandle. *Review of industrial organization* 27, 3 (2005), 253–278.

[14] Julian Faraway. 2006. *Extending the Linear Model with R.* Chapman Hall/CRC Press.

[15] Jeremy Freese and J. Scott Long. 2006. *Regression Models for Categorical Dependent Variables Using Stata.* Stata Press.

[16] Getintent. 2017. RTB Auctions: Fair Play? *AdExchanger.* https://blog.getintent.com/rtb-auctions-fair-play-3b372d505089

[17] Google. 2018. Form 10K for Alphabet Inc. *United States Securities and Exchange Commission.* https://abc.xyz/investor/static/pdf/20180204_alphabet_10K.pdf?cache=11336e3

[18] Google. 2019. Real Time Bidding Protocol Protocol Buffer v.167. *Google Website.* https://developers.google.com/authorized-buyers/rtb/downloads/realtime-bidding-proto

[19] Google. 2019. Real Time Bidding Protocol, Release Notes. *Google Website.* https://developers.google.com/authorized-buyers/rtb/relnotes#updates-2019-03-13

[20] Google. 2020. Real-Time Bidding Protocol Buffer v.173. *Google website.* https://developers.google.com/authorized-buyers/rtb/downloads/realtime-bidding-proto

[21] Kenneth Hendricks and Robert H Porter. 1989. Collusion in auctions. *Annales d'Economie et de Statistique* (1989), 217–230.

[22] Ali Hortaçsu, Jakub Kastl, and Allen Zhang. 2018. Bid shading and bidder surplus in the us treasury auction system. *American Economic Review* 108, 1 (2018), 147–69.

[23] B. Hovaness. 2018. Sold for more than you should have paid. *Hearts and Science Website.* https://www.hearts-science.com/sold-for-more-than-you-should-have-paid/

[24] Niklas Karlsson and Qian Sang. [n.d.]. Adaptive bid shading optimization of first price ad inventory. In *submitted to the 59th IEEE Conf. on Decision and Control.*

[25] Yvan Lengwiler and Elmar Wolfstetter. 2010. Auctions and corruption: An analysis of bid rigging by a corrupt auctioneer. *Journal of Economic Dynamics and control* 34, 10 (2010), 1872–1892.

[26] Rachel AJ Pownall and Leonard Wolk. 2013. Bidding behavior and experience in internet auctions. *European Economic Review* 61 (2013), 14–27.

[27] Rubicon. 2018. Bridging the Gap to First-Price Auctions: A Buyer's Guide. *Rubicon Website.* http://go.rubiconproject.com/rs/958-XBX-033/images/Buyers_Guide_to_First_Price_Rubicon_Project.pdf

[28] Rubicon. 2018. Principles for a Better Programmatic Marketplace, Open Letter from Rubicon, SpotX, OpenX, Pubmatic, Telaris, and Sovr. *Rubicon Website.* https://rubiconproject.com/insights/thought-leadership/principles-better-programmatic-marketplace-open-letter-advertisers-publishers/

[29] Cosma Shalizi. [n.d.]. Logistic Regression Lecture 12. *Undergraduate Advanced Data Analysis 36-402 Lecture Notes* ([n. d.]). https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf

[30] R. Shields. 2019. Google Ad Manager to Offer First-Price Auctions, Simplifying Programmatic Buying. *AdWeek.* https://www.adweek.com/programmatic/google-ad-manager-to-offer-first-price-auctions-simplifying-programmatic-buying//

[31] S. Sluis. 2017. Explainer: More On The Widespread Fee Practice Behind The Guardian's Lawsuit Vs. Rubicon Project. *AdExchanger.* https://adexchanger.com/ad-exchange-news/explainer-widespread-fee-practice-behind-guardians-lawsuit-vs-rubicon-project/

[32] S. Sluis. 2019. Google Switches To First-Price Auction. *AdExchanger.* https://www.adweek.com/programmatic/google-ad-manager-to-offer-first-price-auctions-simplifying-programmatic-buying//

[33] Wush Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2018. Deep censored learning of the winning price in the real time bidding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2526–2535.

[34] Wush Chi-Hsuan Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2015. Predicting winning price in real time bidding with censored data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1305–1314.

[35] Christine Zulehner. 2009. Bidding behavior in sequential cattle auctions. *International Journal of Industrial Organization* 27, 1 (2009), 33–42.