MEOW: A Space-Efficient Non-Parametric Bid Shading Algorithm

Wei Zhang[1,*], Brendan Kitts[2], Yanjun Han[3], Zhengyuan Zhou[4], Tingyu Mao[2], Hao He[2], Shengjun Pan[2], Aaron Flores[2], San Gultekin[2], Tsachy Weissman[3]
[1]Tsinghua University, [2]Yahoo Research, [3]Stanford University, [4]New York University
zhang-w17@mails.tsingua.edu.cn,{yanjun,zyzhou,tsachy}@stanford.edu
{brendan.kitts,aaron.flores}@verizonmedia.com

ABSTRACT

Bid Shading has become increasingly important in Online Advertising, with a large amount of commercial [3, 13, 14, 31] and research work [12, 22, 30] recently published.

Most approaches for solving the bid shading problem involve estimating the probability of win distribution, and then maximizing surplus [30]. These generally use parametric assumptions for the distribution, and there has been some discussion as to whether Log-Normal, Gamma, Beta, or other distributions are most effective [7, 36, 43, 44].

In this paper, we show evidence that online auctions generally diverge in interesting ways from classic distributions. In particular, real auctions generally exhibit significant structure, due to the way that humans set up campaigns and inventory floor prices [8, 17].

Using these insights, we present a Non-Parametric method for Bid Shading which enables the exploitation of this deep structure. The algorithm has low time and space complexity, and is designed to operate within the challenging millisecond Service Level Agreements of Real-Time Bid Servers. We deploy it in one of the largest Demand Side Platforms in the United States, and show that it reliably out-performs best in class Parametric benchmarks. We conclude by suggesting some ways that the best aspects of Parametric and Non-Parametric approaches could be combined.

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:
Wei Zhang[1,*], Brendan Kitts[2], Yanjun Han[3], Zhengyuan Zhou[4], Tingyu Mao[2],, Hao He[2], Shengjun Pan[2], Aaron Flores[2], San Gultekin[2], Tsachy Weissman[3]. 2020. MEOW: A Space-Efficient Non-Parametric Bid Shading Algorithm. In *KDD '20: SIG Conference on Knowledge Discovery and Data Mining, August 2020, San Diego, California, USA.* ACM, New York, NY, USA, 9 pages.

*Corresponding author.

KDD '20, August 2020, San Diego, California, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

1 INTRODUCTION

Between 2017 and 2019, the Online Advertising industry underwent a massive transformation. Prior to 2017, Display ads were sold almost exclusively on Second Price Auctions. However, by 2018, First Price Auctions had increased from 5% to 43% of all auctions [11, 25]. After Google's decision to shift in 2019, 85% of display impressions were sold via First Price.

First Price Auctions present a formidable challenge to advertisers, as they require the bidder to engage in a practice called *bid shading*. Bid shading occurs when the bidder takes the private value that they would have submitted in a second price auction, and then tries to lower their bid so that it is just above the highest competing bid - this in order to minimize their price paid whilst still winning the auction [18]. This process is beset with risk, since the other bids on the auction are unknown. If the bidder shades too little, they will overpay; If they shade too much, they will lose and gain no value. Identifying the optimum bid shade, therefore, requires the bidder to *predict* competing bidder prices, yet without being able to see bidder prices. This is an enormously difficult data mining prediction problem.

With its sudden financial impact in the Online Advertising industry, Bid Shading has become a major area of new research and commercial activity. Several companies announced new bid shading services to help advertisers effectively bid, including Google [13, 14], AppNexus [3] and Rubicon [31]. Researchers and industry practitioners have also published details of new bid shading algorithms [12, 22, 30].

Most approaches for solving the problem involve predicting the probability of winning at different bid prices, the surplus given this probability of win, and then returning the shaded bid price with the maximum surplus [30]. This has generally been accomplished with parametric assumptions on the shape of the landscape.

We show new evidence in this paper that online auctions diverge significantly from well-known distributions, and exhibit usable structure due to the way that humans set up campaigns and inventory floor prices [8, 17]. Parametric distributions often fail to capture this structure.

The current paper presents a new Non-Parametric method for Bid Shading which enables the exploitation of this deep structure. The paper is organized as follows: Section 1 introduces the Bid Shading Problem and describes prior work including several parametric approaches that have been recently published. Section 2 introduces real auction data, and shows that it exhibits significant "spike" structure that not unlike the Left Digit Anchoring Effect observed in Psychology [17]. Section 4 introduces the MEOW algorithm, and proves its time and space complexity. Section 6 shows Offline and Live Bid Server experiments using the new algorithm.

1.1 The Bid Shading Problem

At time t, given a valuation V_t, if we won the impression, 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_t is an accurate representation of the dollar value that the advertiser expects to obtain, and the bid b_t is also in real dollars, the advertiser's financial gain, or *surplus* over a horizon T, is equal to:

$$S_T \stackrel{\text{def}}{=} \sum_{t=1}^{T} (V_t - b_t)\, \mathbf{I}(b_t > m_t), \tag{1}$$

where b_t is the shaded bid, m_t is the minimum bid price to win, and $\mathbf{I}(b_t > m_t) = 1$ if the impression is won, and 0 otherwise. The task is to adaptively find shaded bids $\{b_t\}_{t=1}^{T}$ that maximizes the surplus S_T to the advertiser.

1.2 Related Work

Bid shading is a common tactic in repeated First Price Auctions, and is expected by auction theory. [46] found robust evidence of shading in Austrian livestock auctions, [6] reported shading in a Texas cattle market, and [18] found the practice in auctions for US Treasury notes. Bid Shading shares characteristics with the Seller's (Reserve) Price Optimization Problem [5, 24, 26, 27, 32, 33], although the Bid Shading Problem is a buyer side problem, and the buyer has access to distinct forms of feedback.

A variety of approaches have been proposed to solve the Bid Shading problem:

Winning price predictors There have been some published research on the problem of winning price prediction on auctions. [21, 41, 42] both develop methods for this purpose. Whilst these methods are useful, Bid shading involves predicting and maximizing expected surplus, however, which involves another unknown and optimization step.

Point Estimators use a machine learning algorithm to predict the exact optimal shading factor, by estimating the ratio of the minimum bid to win over the advertiser's private value. For instance, [12] used a Factorization Machine to predict a shading factor by training against known cases of optimum shading factor (0.1).

Unfortunately, this technique is only feasible on Seller auctions in which the Seller provides the exact winning price back to the Buyer (an "Open Bid Auction"). The optimal price can then be used as a training signal for the Buyer. However, most Online Advertising Auction Sellers (including Index Exchange, Pubmatic and others) do not disclose this information, instead just providing whether the bid was accepted or not ("Sealed Bid Auctions"). The "optimum" bid is therefore unknown.

Distribution estimators improve upon these earlier approaches, by training on the 0-1 win/loss signal, and predicting the probability of win across all possible bid prices; effectively creating a probability distribution of winning prices. Once the cdf is predicted, it is possible to calculate the expected surplus at each bid price, and the optimum bid price can be identified.

The recent WinRate model from [30] takes this approach, estimating the win probability for each bid price using a 0-1 logistic distribution, and then maximizing the resulting surplus function. Because of the known parametric form, the authors are able to bound the optimum and use a guaranteed $O(log(K))$ bisection search to find the surplus maximum - the logarithmic time search being highly desirable for Bidding Servers which need to minimize computational operations.

Other authors have taken a similar approach to distribution estimation. [36] extend the Winrate idea to support a range of parametric distribution, and use a Deep Neural Network to estimate the distribution's shape parameters. Their implementation was shown to work for Gamma, Gaussian and Log-Normal distributions. They used a distribution-agnostic, Golden Section Search, to find the surplus maximum bid price.

Although these methods have proven effective, we show in this paper that the actual auction distributions are generally quite divergent from the parametric assumptions, and that more surplus can be captured by modeling the auction data more closely, possibly in a nonparametric way. The usage of nonparametric methods also has a long history in the study of auction theory and first-price auctions. For example, kernel-based methods have been proposed for the nonparametric estimation of bidders' bid distribution [15], or the value distribution [35], sometimes with an unknown number of bidders [2], in first-price auctions. However, this thread of research typically assumes strong assumptions on the bid distribution such as a smooth density, which usually does not hold in real bidding data. These works are in sharp contrast with the recent work [16], which relies on few modeling assumptions and is the building block of the MEOW algorithm proposed in this paper.

2 AUCTION LANDSCAPES

In auction literature and past work, Auction State bids are often assumed to be normally distributed [43, 44]; although others have noted that their auction data was fit well by Log-Normal [7]. We tested a variety of distributions on our auction data. None of the distributions fit well enough to be significant under a test for fit, but similar to [44] we found that Log-Normal has the lowest error and highest linear correlation to the actual data, out of Normal, Log-Normal and Gamma distributions tested. Log-Normal is seen in other auctions such as contract bidding [34].

In addition to the overall shape, Auction State also has some unusual characteristics. One that stands out is that there are some prices where there are large spikes in impressions. For example, Figure 1 is typical.

Some previous authors address the spikes by smoothing over them. [23] fit smoothing splines to the bid-volume cumulative distribution. [45] use a Mixture of Gaussians. If the spikes are noise, then avoiding transient spikes should result in better out-of-sample prediction. But what if the spikes are not actually noise?

Spikes appear in other domains: Marathon running times are approximately Log-normal. However, finishing times also have spikes on the left-side of hours and half hours. This is due to a

Pennies	Online Auction Bids	Internet prices [2]	Retail prices [3]
00..09	27%	28%	8%
10..19	9%	3%	0%
20..29	8%	4%	0%
30..39	8%	4%	1%
40..49	7%	4%	0%
50..59	18%	4%	29%
60..69	6%	3%	0%
70..79	6%	4%	0%
80..89	6%	6%	1%
90..99	6%	41%	61%

Table 1: End Digit distributions. "Online Auction Bids" are from observed Verizon SSP ad prices

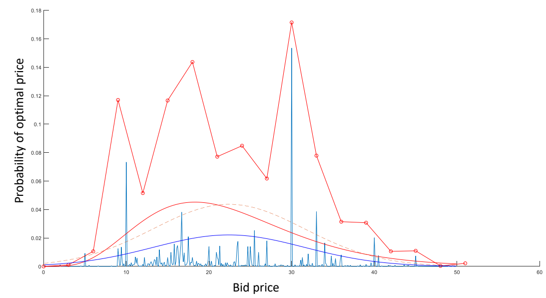

Figure 1: Online auctions exhibit unusual distributions of bid prices. In the above example, spikes occur at certain bid prices. This results in Normal, Log-Normal and Gamma distributions providing poor fits to this data. Furthermore attempts to smooth the spikes actually decreases predictive power. The spikes appear to be real phenomena due to human pricing effects

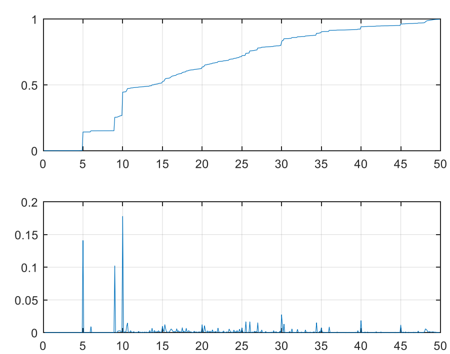

Figure 2: Marginal distribution (bottom) versus cumulative distribution (top) spikes in price can be clearly seen at 5, 9.5, 10, 15, 20, 25, 30, 35, 40, 45. These price spikes seem to be related to the Psychology of price setting on fixed price contracts.

Psychological effect where humans favor finishing before whole hours and half hours [10, 39].

Spikes also occur in pricing. A histogram of retail prices usually shows spikes at price ending in 9 and 5. This effect is known as the "Left Digit Anchoring Effect", a Psychological phenomenon where consumers seem to ignore the least significant right-hand-side digits when doing value comparisons [17]. A range of theories have been offered to explain the practice, including cognitive workload from rounding up, precision being taken as an indicator of truthfulness, and others. These unusual price-points are a robust part of retail price optimization data.

In the field of online advertising, website owners set floor prices for inventory, which, in turn, impact the auction prices we observe. We encounter something unusual: The floors primarily use round numbered prices, including 5, 9.50, 10, 15, 20, 25, 30, 35, 40, 45. This can be seen clearly in Figure 2. It looks like Supply managers are susceptible to an "End Digit Effect" also!

Table 1 shows a comparison of End Digit Effects in other domains. It therefore matters whether a price of $10 or $9.99 is submitted to the auction – there really are "cliffs" in terms of impressions at different round price thresholds.

In order to capture these human-engineered artifacts, we need more freedom to model the data. Jacob Wolfowitz introduced the term *Non-parametric*, in 1942, as a way of describing methods that did not rely on data belonging to any particular parametric family of probability distributions [40]. Estimating the auction surplus cdf by discretizing and estimating regions separately - is certainly a Non-parametric approach [1, 28, 37] and might offer a way of modeling the unusual structure in auctions.

3 PROBLEMS WITH NON-PARAMETRIC ALGORITHMS

Non-parametric methods present a range of challenges which need to be resolved before they can be used.

One challenge is high storage cost. A parametric model of an auction surplus cdf will only require $O(p)$ parameters, where p is the number of features being used. A discrete approximation of the same cdf, will require $O(p \times K \times M)$ where K is the number of possible bids and M is the number of private valuations. If K and M are in units of CPM with the smallest unit of bid a penny, and span all 2 place numeric values greater than 0; for bids between $0 and $10, and valuations between $0 and $100; that means $1{,}000 \times 10{,}000 = 10$ million bins.

A second challenge is generalization. Figure 3 shows performance of a Fixed-width Non-parametric algorithm, versus the average historical winning price. The Non-Parametric algorithm, in this example, works best when there are more than 80,000 auction observations. However, below this threshold, the Non-Parametric approach actually *performs worse* than the simple strategy of predicting the mean. The reason for this loss of performance, is because the Non-parametric algorithm's binning is too fine-grained, resulting in sparse data which doesn't carry statistical significance. Thus,

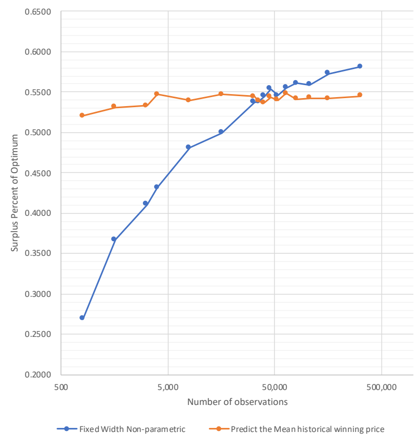

Figure 3: Fixed width non-parametric predictors can perform worse than the mean if their resolution is too high for the data.

bid \ valuation	v_1	v_2	$\cdots$	v_j	$\cdots$	v_M
b_1				$S_{1,j}$		
b_2				$S_{2,j}$		
$\vdots$				$\vdots$		
b_i				$S_{i,j}$		
$\vdots$				$\vdots$		
b_K				$S_{K,j}$		

Table 2: A reward table used in the SEW algorithm.

the Non-parametric algorithm needs the ability to dynamically adjust its bin sizes, so that it maintains usable resolution. The MEOW algorithm, described below, uses dynamic binning.

4 MEOW ALGORITHM

To mitigate the issues of storage cost and generalization, we introduce a nonparametric algorithm called the Multi-resolutional Exponential Weighting (ME(O)W) algorithm. This algorithm inherits the nice theoretical properties of the Exponential Weighting based algorithm [16] through the lens of online learning, and uses a dynamic and data-driven binning to significantly reduce the memory requirement and adapt to different natures of data.

4.1 Algorithm Overview

The MEOW algorithm is motivated by the general idea of exponential weighting in nonparametric bid shading, where both the private values and bidding prices are quantized into discrete levels, and we maintain a table of historic rewards with each entry corresponding to a given pair of private value and bidding price. At each time, the private value is computed, and the bidder's bid is determined by running an exponential weighting algorithm on the rewards of all candidate prices given this private value. An example of the reward table is illustrated in Table 2, where $S_{i,j}$ denotes the cumulative historic surplus of bidding b_i under the private valuation v_j. Under the private value v_j, the exponential weighting algorithm selects a random bid b_i with probability

$$p_i = \frac{\exp(\eta S_{i,j})}{\sum_{k=1}^{K} \exp(\eta S_{k,j})},$$

where $\eta > 0$ is a properly chosen learning rate.

However, maintaining such a static table is typically very memory-consuming, leaving lots of bins seldom visited, and a large portion of candidate prices probably too bad for the bidder to bid. Also, the non-data-driven nature of the table leads to a poor generalization performance. The MEOW algorithm improves over the static table by choosing its rows and columns in a dynamic and data-driven way, and specifically greatly reduces the quantization levels for both the private value (horizontally) and bidding prices (vertically).

Horizontal: private value bins. The high-level idea of horizontal binning is to adapt the bin design to the real data distribution, where each bin has comparatively similar amounts of data. Specifically, if some bin of private values consists of too much data, we further split it into smaller bins to reduce the quantization error. On the other hand, if some bin has too little data, we merge it into another bin so that there is enough data in this bin for learning. In the MEOW algorithm, we first fix a static binning, and then perform the splitting and merging operations of bins based on incoming data. To reduce the computational cost, in the algorithm these steps are only performed every T_1 rounds of auctions, where $T_1 > 0$ is a hyperparameter which is moderately large (e.g. $T_1 = 1,000$).

Vertical: bidding price levels. The redundancy in the price levels comes from the fact that, the optimal bidding price given a private value in a small bin also lies in a small range. Therefore, we could roughly estimates the optimal price (possibly with a low precision) and then keep only a few candidate prices around it. Specifically, for each bin of private values, we use the historic data to compute the empirically optimal bidding price p^* in this bin, and the set of candidate bidding prices is chosen to be a suitable quantization of $[p^* - \Delta, p^* + \Delta]$ for some small $\Delta > 0$. The final quantization level could be as small as $5 \sim 20$, which greatly reduces the storage cost. In the MEOW algorithm, the process of updating candidate bidding prices is implemented every T_2 rounds, where $T_2 > 0$ is a relatively long time (e.g. 1 day).

Discount factor. The final MEOW algorithm also involves a discount factor $\sigma \in (0, 1)$ for two purposes. First, practical data are typically non-stationary over time, and gradually forgetting old data enables a better adaptation to the new data. Second, as the amount of data increases, without the discount factor it is possible to have infinite bins of private values, which increases both the computational and the storage cost. In contrast, with a discount factor, the number of bins is always bounded from above (cf. Theorem 1). In the MEOW algorithm, we will apply this discount factor to both the data counts of each bin, and also the cumulative reward of each candidate bidding price.

Algorithm 1: Multi-resolutional Exponential Weighting (MEOW)

```
Inputs: Initial number of bins M_0; Initial ranges V, P;
  Number of prices K; Discount factor σ ∈ (0, 1); Learning
  rate η > 0; Update periods T_1, T_2; Thresholds N_1, N_2.
Initialization: Build M_0 bins equally for v_t ∈ [0, V], and
  set bin.price[j] = jP/K for each j = 1, ···, K.
for t = 1, 2, ... do
    % Search for current bin
    Observe private value v_t;
    if v_t > V then
        Create a new bin [floor(v_t), floor(v_t) + 1);
    end
    Search for the bin* s.t. v_t ∈ [bin*.v_low, bin*.v_high);
    % Exponential weighting
    for j = 1, 2, ... K do
        prob[j] ← exp(η · bin*.history[j])
    end
    Sample b_t ~ prob/Σ_{j=1}^{K} prob[j];
    % Bin update
    Observe the minimum-bid-to-win m_t;
    for j = 1, 2, ... K do
        bin*.history[j] ← bin*.history[j] +
          instantreward(bin*.price[j]; v_t, m_t);
    end
    bin*.count ← bin*.count + 1;
    % Split or merge bins after every T_1 steps
    if t%T_1 == 0 then
        for all bins do
            % Split a large bin into two smaller bins
            if bin.count ≥ N_1 then
                bin.history ← bin.history/2;
                bin.count ← bin.count/2;
                Create new bins bin_l, bin_r ← bin;
                bin_l.v_high ← (bin.v_low + bin.v_high)/2;
                bin_r.v_low ← bin_l.v_high;
                Replace bin by bin_l and bin_r;
            end
            % Merge two smaller bins into a large bin
            if bin.count ≤ N_2 then
                Find the neighbor bin′ with a smaller count;
                Create a new bin* with private value range
                  [bin.v_low, bin.v_high)∪[bin′.v_low, bin′.v_high);
                bin*.count ← bin.count + bin′.count;
                bin*.price and bin*.history inherit from the
                  bin with a larger count;
                Remove bin and bin′, and add bin*;
            end
        end
        % Discount
        Multiply all counts and histories by the factor σ;
    end
    % Update price levels after every T_2 steps
    if t%T_2 == 0 then
        Requantization();
    end
end
```

Algorithm 2: Requantization

```
Global inputs: private value bins, number of price levels K
for all possible bin do
    j* ← argmax(bin.history);
    for j = 1, 2, ... K do
        bin.price[j] ← bin.price[j* − 7] + j ·
          (bin.price[j* + 7] − bin.price[j* − 7])/K;
    end
    bin.history ← 0;
end
```

4.2 Algorithm Details

The complete description of the MEOW algorithm is summarized in Algorithm 1, which also takes Algorithm 2 as a subroutine which updates the candidate bidding price every T_2 time steps. Specifically, the MEOW algorithm maintains an array of private value bins, where each bin is a data structure consisting of the following variables:

- $[v_{\text{low}}, v_{\text{high}})$: range of the private value in the bin;
- count: cumulative (discounted) amount of past data falling into this bin;
- price$[K]$: an array of K candidate bidding prices under this bin, sorted in an increasing order;
- history$[K]$: an array of K cumulative (discounted) historic rewards associated with the above K bidding prices.

Here $K > 0$ is a fixed parameter in the algorithm and denotes the number of vertical quantization levels.

In the initialization of the algorithm, we uniformly partition the interval $[0, V]$ into M_0 bins, where $V > 0$ is an upper bound for most private values (e.g. the 1% quantile). For each bin, we initialize K price levels to be a uniform quantization of $[0, P]$, where $P > 0$ is the maximum bidding price. All the counts and the reward histories are initialized to be zero.

Next we describe the dynamic updates of the private values and candidate prices, respectively. For the private value bins, if some private value above V occurs (which is unlikely), we create a new bin for this value. After every T_1 time steps, we check the size of each bin: if the bin count is larger than a threshold N_1, we split it evenly into two bins, with both the count and history halved; if the bin count is smaller than another threshold N_2, we merge it with one of its neighboring bin which a smaller size, combine their counts, and inherit the prive levels and history from the larger bin. We repeat this process until the count of each bin is between $[N_2, N_1]$, and then apply the discount factor $\sigma \in (0, 1)$ to both the counts and the historic rewards.

As for the updates of candidate prices, we call the subroutine in Algorithm 2 every T_2 time steps. In Algorithm 2, for each bin we pick the best price p^* giving the largest historic reward, and update the new prices to a uniform K-level quantization of $[p^* - \Delta, p^* + \Delta]$ for some Δ. The specific choice of the interval is based on the past price level: the algorithm finds the best bidding price bid.price$[j^*]$, and chooses the interval to be $[\text{bid.price}[j^* - 7], \text{bid.price}[j^* + 7]]$. Finally, since the prive levels have changed, we also reset all historic rewards to zero.

Finally we provide an example choice of the hyperparameters used in our experiments: $M_0 = 40, V = 100, P = 10, K = 20, \sigma = 0.99, \eta = 1, (T_1, T_2) = (1000, 1 \text{ day}), (N_1, N_2) = (2500, 10000)$. Note that we will ignore the first few T_1 steps to ensure enough data for splitting/merging when we restart the algorithm.

4.3 Time and Space Complexity

In this section we provide the analysis on the space and tiem complexities of the MEOW algorithm, and show that it could indeed be efficiently implemented in practice. We start by showing that thanks to the discount factor, the number of bins is always finite.

Theorem 1. *Even for an infinite amount of data, the total number of bins is upper bounded by a constant number*

$$M := \max\left\{\frac{T_1}{N_2 \cdot (1-\sigma)}, M_0\right\}.$$

Proof. First we show that due to the discount, the total count is bounded by a constant value:

$$\sum_{\text{all possible bins}} (\text{bin.count}) \le T_1 \cdot (1 + \sigma + \sigma^2 + \cdots) < \frac{T_1}{1-\sigma}.$$

Since after each horizontal bin update, the count of each bin is at least N_2. In view of the above inequality, the number of bins after update is at most $T_1/(N_2(1-\sigma))$. Moreover, before all bin updates the number of bins is initialized to be M_0, and the result follows. □

Under the choice of parameters $T_1 = 1000, N_2 = 2000, \sigma = 0.99$, and $M_0 = 40$, we compute that $M = 40$ in Theorem 1. Consequently, our storage cost is at most $O(MK)$, corresponding to the storage of the matrix consisting of all historic rewards.

As for the computational complexity, note that whenever there is no horizontal or vertical update, the running time of the bin search and the exponential weighted prediction in Algorithm 1 is at most $O(\log M + K)$. When there is either a horizontal or a vertical update, we may need to change the history table for all bins, which takes $O(MK)$ time. Therefore, the overall time complexity during T rounds of auctions is

$$O\left(T \cdot \left(\log M + K + \frac{MK}{T_1} + \frac{MK}{T_2}\right)\right),$$

which is linear in T with the coefficient smaller than 30 under our parameter configuration.

5 IMPLEMENTATION

The bid shading system was deployed on Verizon Demand Side Platform (VZDSP) [29], the fourth largest in the United States after Google, Amazon, and the Trade Desk [9]. The performance requirements for VZDSP are extreme. At run-time, the Bid Shader needs to respond to 5.5 million requests per second peak load. For each of these bid requests, a bid needs to be calculated within 100 miliseconds. Approximately 1000 bid servers are used to serve ads, which means that each server has to handle 5,000 requests per second. Overall, less than 10 miliseconds are budgeted for bid calculations. The time complexity of Section 4.3 shows that the algorithm only adds about 30 additional operations per request.

Space requirements are also highly restrictive. Bid servers carry about 28 Gigabytes of RAM. There are over 200,000 sub-domains and mobile applications requesting bids. Therefore assuming M=40 and K=20, there are $200{,}000 \times M \times K$ = 160,000,000 = 160 million double types are needed, which equals about 1.28 Gigabytes RAM per bid-server. The analysis to follow shows results for the most frequent 100 domains, the memory consumption for which was negligable at 0.64 Megabytes.

6 EXPERIMENTS

In order to measure the performance of the bid shading system, two forms of testing were performed: (i) Leveraging knowledge of the highest competing bids provided by an ad exchange, auctions were replayed using the MEOW algorithm to calculate bid prices. (ii) the MEOW algorithm was also used in production in an A/B test vs the production algorithm. The production algorithm benchmark in both cases was an implementation of log-normal distribution-based shading [30].

6.1 Offline Auction Replay

The Non-parametric algorithm was first tested on saved auction data captured from the Verizon Demand Side Platform where for each auction/bid request, private valuations and highest competing bids were known. Bid requests from the top 100 top-level domains were used (cnn.com, espn.com, buzzfeed.com and other sites), and all auctions from December 22 to January 12 2021 in which the Production algorithm responded with a bid, were used. This comprised approximately 6.2 billion requests.

The data spanned an interesting period of time, since it ranged from the 2020 Christmas shopping season with high advertising prices around \$0.97 CPM, to January 2021 in which advertising prices dropped to just \$0.77 CPM.

Figure 4 shows the behavior across this period; after good performance from December 22 to 29, there's a big drop centered on January 1, 2021. Surplus yield worsens by 15% due to the new auctions no longer matching historical data. On January 2nd, the surplus recovers. Quantitatively, MEOW appears to have responded better to the change in distribution over the dataset. The R^2 between algorithm bid and optimal bid for Distribution was 0.903 where-as for MEOW it was 0.953. The mean absolute difference in CPM was \$0.88 and \$0.64; and surplus as a percent of optimal surplus was 47.9% and 53.4% for Distribution vs MEOW (Table 3). Thus MEOW submitted bids that were closer to optimum and had a better correlation in matching the in-time changes to the optimum bid distribution.

Table 4 summarizes the performance on replay data. The increase in surplus ranged between 5.7% (100th percentile) to 10.1% (90th percentile), and all increases were statistically significant ($p<0.01$; paired t-test; MEOW vs Distribution surplus scores compared daily). The table reports on several surplus percentiles because we have found it to be common for a tiny percentage of advertiser ads to have unrealistic goals and be "chronically wound up" by the control system, resulting in spuriously high surpluses. Therefore we presented a range of percentiles from 90% to 100% to help verify that the results were robust.

Metric	Distr.	EW
surplus % of opt	47.91%	53.42%
imps % of opt	48.01%	61.21%
spend % of opt	43.03%	62.28%

Table 3: Optimality

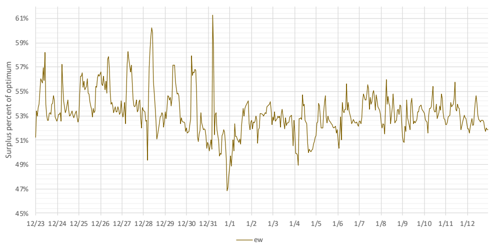

Figure 4: Surplus captured by Non-parametric algorithm, as a percent of optimal surplus from December 22 to January 12, 2021

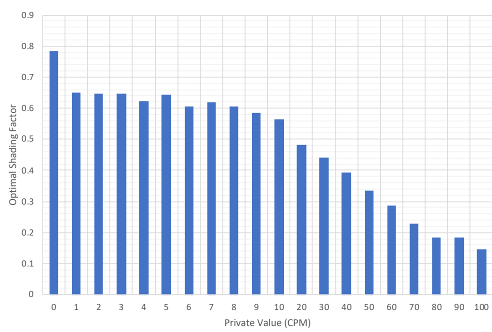

Figure 5: Predicted Optimal Shading Factor by Private Value average over all domains

6.2 Online Production Performance

The algorithm was also deployed in the Verizon DSP bid server [29], and was set to run on a randomly selected 1 percent of traffic, and the top 100 domains. The period of data analyzed spans from January 21 to January 29 2021.

The results are shown in Table 5. The surplus increased between 3.3% to 6% (90th..100th percentile; all except the 100th percentile were significant at p<0.01; paired t-test).

6.3 Observations

Some examples of MEOW behavior on real Online Advertising auction data (Offline and Online experiments) are shown in Figures 5, 6, 7.

Figure 7 shows the performance MEOW approximating the actual win cumulative probability distribution for ebay.com in its offline experiment. For each bid price, the MEOW algorithm submits

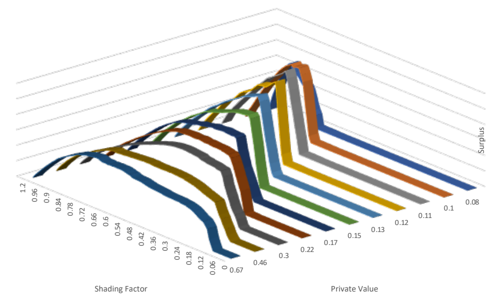

Figure 6: Predicted Surplus given Private Value and Shading Factor for one domain

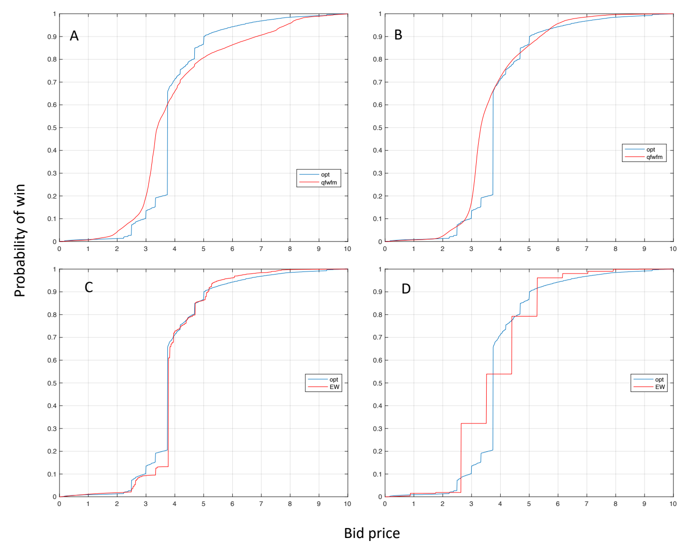

Figure 7: Comparison of probability of win for (A) Point Estimator [12], (B) Distribution [30], (C) MEOW with K=30 bins, and (D) MEOW with K=7 bins.

percentile	90	95	98	99	100
mean	10.1%	11.5%	10.4%	8.7%	5.7%
stdev	4.4%	4.1%	3.6%	3.3%	1.8%
stderr	1.0%	0.9%	0.8%	0.7%	0.4%
ttest	<0.001	<0.001	<0.001	<0.001	<0.001

Table 4: Surplus Offline Results

its predicted optimal bid. We have aggregated those bid submissions into a cumulative probability distribution. We then compared this against the actual win probability distribution in the data.

The Distribution based approach [30] does a good job of approximating the optimal bid price distribution. However there are clear imperfections in high bids; indeed we sometimes find that the

percentile	90	95	98	99	100
mean	3.3%	6.1%	4.5%	3.5%	6.0%
stdev	2.7%	3.3%	4.8%	3.9%	22.0%
stderr	0.72%	0.89%	1.28%	1.03%	5.87%
ttest	<0.001	<0.001	0.004	0.003	0.22

Table 5: Surplus Online Results

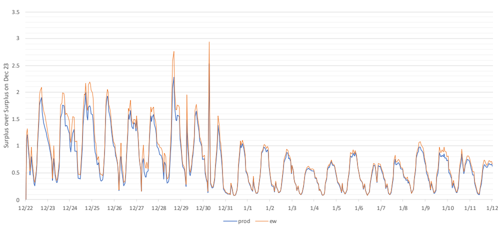

Figure 8: Offline MEOW versus Distribution Bid Shading algorithms Dec 22 - Jan 12 2021. Bid prices decrease significantly starting January 1

Distribution estimates are biased systematically due to the shape that they are required to fit. We also observe that the Distribution approach has difficulty setting the win probability to zero for the "floor" price of the auction - instead of an immediate drop to zero, it is a gentle slope.

For illustration purposes, we show a "low resolution" MEOW algorithm that only has K=7 bins; the algorithm approximation is relatively poor. In contrast, the MEOW algorithm with K=30 bins approximates the distribution extremely well - and better than the Distribution approach. In particular, the Non-parametric approach approximates the floor (bid prices below $2.00) and the ceiling.

Figure 5 shows the importance of private value quantization. This is the average of shading factors which associated with the maximum surplus, for each private value bin. As private value increases, the algorithm finds that a deeper shading factor is optimal, a result also observed in the auction literature [4].

Figure 6 shows the relationship between private value, shading factor, and surplus, for one domain (spotify.com). It can be seen that there is a bid price region where the system reliably has zero probability of winning. This is likely the auction floor which the system has inferred.

7 DISCUSSION

The higher yield from Non-parametric algorithms isn't free. Whereas the Distribution algorithm might typically have two numeric parameters for its distribution shape (variance and mean for example), MEOW has $M \times K$ parameters; which for the default parameters of M=40 and K=20 results in 800 doubles. Thus, the algorithm is about 400 times more expensive in space. We've argued, and the experimental results also support, the argument that this additional space is needed to capture the various spike patterns. However, the same level of resourcing might not be necessary for every auction.

We believe it might be possible to combine both parametric and nonparametric approaches, and use the higher precision of non-parametric where needed, and preserve storage when parametric approximates well enough. One approach that seems promising is the *online learning with hints* framework from [38], where the parametric fit forms a hint which is used or discarded based on performance. We believe future work in this area may be fruitful.

8 CONCLUSION

The shift to First Price has been traumatic for the advertising industry. Several researchers reported that traffic prices for First Price Auctions increased between 5% and 50% higher compared to Second Price Auctions [3, 19, 25, 31], meaning significantly lower advertiser profitability for the same impressions. [25] reported that after their SSP switched to First Price, 10% of advertisers actually discontinued bidding.

As a result of these problems, there has been an explosion of research and commercial implementations in Machine Learning for Bid Shading.

It seems certain that the financial imperative to shade better than competing Demand Side companies, will lead bidders to begin to exploit the deep pricing structure in online auctions. There appears to be plenty of performance available, for researchers and companies who are willing to "listen to what their data is telling them".

As Thomas Huxley, the great biologist and supporter of Charles Darwin suggested, we should endeavor to *"..sit down before fact as a little child, be prepared to give up every preconceived notion, follow humbly wherever and to whatever abysses nature leads, or you shall learn nothing."* - Thomas Huxley (1860), [20]

REFERENCES

[1] 2021. Nonparametric statistics. *Wikipedia website* (2021). https://en.wikipedia.org/wiki/Nonparametric_statistics

[2] Yonghong An, Yingyao Hu, and Matthew Shum. 2010. Estimating first-price auctions with an unknown number of bidders: A misclassification approach. *Journal of Econometrics* 157, 2 (2010), 328–341.

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

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

[5] A. Blum and J. D. Hartline. 2005. Near-optimal online auctions. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms* (January 2005), 1156–1163.

[6] 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.

[7] Zhang-R. Li W. Mao J. Cui, Y. [n.d.]. Bid Landscape Forecasting in Online Ad Exchange Marketplace. (21 August [n. d.]).

[8] Haipeng (Allan) Chen Robert J. Kauffman Daniel Levy, Dongwon Lee and Mark Bergen. 2011. Price Points and Price Rigidity. *Review of Economics and Statistics* 93. Issue 4. https://www.biu.ac.il/soc/ec/d_levy/wp/pricepointsaug2008.pdf

[9] emarketer. 2020. Top 6 DSPs in the past 12 months.

[10] Devin G. Pope George Wu Eric J. Allen, Patricia M. Dechow. 2014. Reference-Dependent Preferences: Evidence from Marathon Runner, Management Science. (2014). http://faculty.chicagobooth.edu/devin.pope/research/pdf/website_marathons.pdf

[11] Getintent. 2018. Digital Ad Impression Share Among US Supply-Side Platforms (SSPs), by Auction Type, Dec 2017 and March 2018: % of total impressions analyzed by GetIntent, April 30, 2018. https://www.emarketer.com/content/five-charts-the-state-of-programmatic-bidding

[12] Djordje Gligorijevic, Tian Zhou, Bharatbhushan Shetty, Brendan Kitts, Shengjun Pan, Junwei Pan, and Aaron Flores. 2020. Bid Shading in The Brave New World of First-Price Auctions. In *The 29th ACM International Conference on Information*

and Knowledge Management. CIKM'20.

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

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

[15] Emmanuel Guerre, Isabelle Perrigne, and Quang Vuong. 2000. Optimal nonparametric estimation of first-price auctions. *Econometrica* 68, 3 (2000), 525–574.

[16] Yanjun Han, Zhengyuan Zhou, Aaron Flores, Erik Ordentlich, and Tsachy Weissman. 2020. Learning to Bid Optimally and Efficiently in Adversarial First-price Auctions. *arXiv preprint arXiv:2007.04568* (2020).

[17] Gendall P. Garland R. Holdershaw, J. 1997. The Widespread Use Of Odd Pricing In The Retail Sector. *Marketing Bulletin* 8. http://marketing-bulletin.massey.ac.nz/V8/MB_V8_N1_Holdershaw.pdf

[18] 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.

[19] B. Hovaness. 2018. Sold for more than you should have paid. https://www.hearts-science.com/sold-for-more-than-you-should-have-paid/

[20] Thomas Henry Huxley and Leonard Huxley. 1900. *The Life and Letters of Thomas Henry Huxley*. Vol. 1. Macmillan.

[21] Verizon Media internal report. 2019. Predicting Optimal Bid Shading Factor Using Logistic Regression.

[22] Niklas Karlsson and Qian Sang. 2020. Adaptive bid shading optimization of first price ad inventory. In *submitted to the 59th IEEE Conf. on Decision and Control*. IEEE CDC'20.

[23] N. Karlsson and Zhang. 2017. A Forecasting-Based Novel Feedback Campaign Control System. (2017).

[24] N. B. Keskin and A. Zeevi. 2014. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. *Operations Research* 62, 5 (2014), 1142–1167.

[25] Brendan Kitts. 2019. Bidder Behavior after Shifting from Second to First Price Auctions in Online Advertising. http://www.appliedaisystems.com/papers/FPA_Effects33.pdf

[26] B. Kitts and K. Hetherington-Young. 2005. Price Optimization in Grocery Stores with Cannibalistic Product Interactions. *Proceedings of the First Workshop on Data Mining Case Studies, Fifth IEEE International Conference on Data Mining (ICDM 2005)* (November 2005), 74–91. http://dataminingcasestudies.com/DMCS_WorkshopProceedings25.pdf

[27] R. Kleinberg and T. Leighton. 2003. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. *44th Annual IEEE Symposium on Foundations of Computer Science* (2003), 594–605.

[28] Paul Kvam and Brani Vidakovic. 2007. *Nonparametric Statistics with Applications in Science and Engineering*. John Wiley & Sons, Inc. http://zoe.bme.gatech.edu/~bv20/isye6404/Bank/npmarginal.pdf

[29] Christina MacDonald. 2020. Verizon Media brings native marketplace, premium inventory into expanded DSP. https://www.verizonmedia.com/press/2020/05/01/verizon-media-expanded-dsp

[30] Shengjun Pan, Brendan Kitts, Tian Zhou, Hao He, Bharatbhushan Shetty, Aaron Flores, DjordjeGligorijevic, Junwei Pan, Tingyu Mao, San Gultekin, and Jianlong Zhang. 2020. Bid Shading by Win-Rate Estimation and Surplus Maximization. In *The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. AdKDD'20, San Diego, California, USA.

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

[32] Ilya Segal. 2003. Optimal Pricing Mechanisms with Unknown Demand. *American Economic Review* 93, 3 (June 2003), 509–529. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=297674

[33] Hermann Simon. 1989. Price Management. (August 1989).

[34] Pablo Ballesteros-Pérez & Martin Skitmore. 2017. On the distribution of bids for construction contract auctions. *Construction Management and Economics* 35, 3 (2017), 106–121.

[35] Unjy Song. 2004. *Nonparametric estimation of an eBay auction model with an unknown number of bidders.* Citeseer.

[36] Shengjun Pan Niklas Karlsson Bharatbhushan Shetty Brendan Kitts DjordjeGligorijevic Junwei Pan San Gultekin Tingyu Mao Jianlong Zhang Tian Zhou, Hao He and Aaron Flores. 2021. Efficient Deep Distribution Network for Bid Shading in First Price Auctions. *unpublished, in review* (2021).

[37] L. Wasserman. 2006. *All Things Nonparametric*. Springer Verlag.

[38] Chen-Yu Wei, Haipeng Luo, and Alekh Agarwal. 2020. Taking a hint: How to leverage loss predictors in contextual bandits?. In *Conference on Learning Theory*. PMLR, 3583–3634.

[39] J. Wolfers. 2014. What Good Marathons and Bad Investments Have in Common. *New York Times* (22 April 2014). https://www.nytimes.com/2014/04/23/upshot/what-good-marathons-and-bad-investments-have-in-common.html

[40] J. Wolfowitz. 1942. Additive Partition Functions and a Class of Statistical Hypotheses. *Annals of Statistics* (1942), 247–279.

[41] 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*. KDD '18, London, United Kingdom, 2526–2535.

[42] 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*. KDD '15, Sydney, 1305–1314.

[43] Yeh-M.Y. Chen M.S. Wu, W.C.H. 2015. Predicting winning price in real time bidding with censored data. (2015).

[44] Lee-K. Wang L. Xie, Z. 2017. Optimal Reserve Price for Online Ads Trading Based on Inventory Identification. (13 August 2017).

[45] Guo-J. Karlsson N. Zhou, T. 2018. A KPIs Forecasting Scheme for Online Advertising Systems. (2018).

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