Online Learning for Measuring Incentive Compatibility in Ad Auctions*

Zhe Feng†
Harvard University
Cambridge, MA, USA
zhe_feng@g.harvard.edu

Okke Schrijvers
Facebook Research
Menlo Park, CA, USA
okke@fb.com

Eric Sodomka
Facebook Research
Menlo Park, CA, USA
sodomka@fb.com

ABSTRACT

In this paper we investigate the problem of measuring end-to-end Incentive Compatibility (IC) regret given black-box access to an auction mechanism. Our goal is to 1) compute an estimate for IC regret in an auction, 2) provide a measure of certainty around the estimate of IC regret, and 3) minimize the time it takes to arrive at an accurate estimate. We consider two main problems, with different informational assumptions: In the *advertiser problem* the goal is to measure IC regret for some known valuation v, while in the more general *demand-side platform (DSP) problem* we wish to determine the worst-case IC regret over all possible valuations. The problems are naturally phrased in an online learning model and we design Regret-UCB algorithms for both problems. We give an online learning algorithm where for the advertiser problem the error of determining IC shrinks as $O\left(\frac{|B|}{T}\cdot\left(\frac{\ln T}{n}+\sqrt{\frac{\ln T}{n}}\right)\right)$ (where B is the finite set of bids, T is the number of time steps, and n is number of auctions per time step), and for the DSP problem it shrinks as $O\left(\frac{|B|}{T}\cdot\left(\frac{|B|\ln T}{n}+\sqrt{\frac{|B|\ln T}{n}}\right)\right)$. For the DSP problem, we also consider stronger IC regret estimation and extend our Regret-UCB algorithm to achieve better IC regret error. We validate the theoretical results using simulations with Generalized Second Price (GSP) auctions, which are known to not be incentive compatible and thus have strictly positive IC regret.

CCS CONCEPTS

• **Theory of computation → Online learning theory**; **Algorithmic game theory**.

KEYWORDS

Ad Auctions; Incentive Compatibility; Multi-armed Bandits

ACM Reference Format:
Zhe Feng, Okke Schrijvers, and Eric Sodomka. 2019. Online Learning for Measuring Incentive Compatibility in Ad Auctions. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313674

*The full version of this paper, including the proof details, is available on ArXiv.
†This work was done while the first author was an intern in Core Data Science, Facebook Research.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313674

1 INTRODUCTION

Online advertising has grown into a massive industry, both in terms of ads volume as well as the complexity of the ecosystem. Most inventory is sold through auctions, and demand-side platforms (DSPs)—such as AppNexus, Quantcast, Google's DoubleClick and Facebook's Audience Network—offer advertisers the opportunity buy inventory from many different publishers. In turn, publishers themselves may run their own auctions to determine which ad to show and what price to charge, resulting in a complicated sequence of auction systems. While auction theory has influenced the design of online ad auctions, in practice not all auctions are incentive compatible. Practices such as sequential auctions (where the winners of an auction on a DSP compete in further auctions on the publisher website) and "dynamic reserve prices" (where publishers use a reserve price of e.g. 50% of the highest bid in the auction) both violate incentive compatibility, even when individual auctions are otherwise designed to be incentive compatible.

A lack of incentive compatibility in the ecosystem is problematic both for advertisers and DSPs. Advertisers have to spend time and energy in figuring out the optimal bidding strategy for different placements, a process which can be quite costly especially when incentives differ across different publishers. For DSPs this is problematic because common budget management techniques, such as multiplicative pacing, are only optimal when the individual auctions are incentive compatible [10].

Since (the lack of) incentive compatibility impacts bidding strategies for advertisers and the quality of the product that a DSP offers, it's important to understand the end-to-end incentive compatibility of an ad system. But how should we think of quantifying incentive compatibility? IC is a binary property: either a buyer maximizes their utility by bidding truthfully or she doesn't. However, this fails to capture that a first-price auction is "worse," in some sense, than a second-price auction with a dynamic reserve price of 50% of the highest bid. To capture these differences, we focus on dynamically measuring *IC regret*:

$$\text{IC regret}(v_i) = \max_{b_i} \mathbb{E}_{b_{-i}}\left[u_i(b_i, b_{-i}) - u_i(v_i, b_{-i})\right], \qquad (1)$$

where v_i is the true value of advertiser i, b_i the bid of i, b_{-i} the bids of other advertisers, and $u_i(\cdot)$ the (expected) utility of i. IC regret captures the difference in utility between bidding truthfully, and the maximum utility achievable. By definition, incentive compatible mechanisms have IC regret 0, while higher IC regret indicates a stronger incentive to misreport.

In this paper we focus on measuring IC regret assuming only black-box access to the auction mechanism. That is, for a given advertiser bid b_i we observe whether this yielded an impression x_i, and what price p_i was paid if any. While it seems restrictive to only focus on black box access, this setting allows one to deal with the fact that auction logic is spread out over different participants in the ecosystem (e.g. a DSP and publisher may each run an auction using proprietary code), and it even allows an advertiser who has no access to any auction code to verify the end-to-end incentives in the system.

Given black-box access to the auction mechanism, our goal is to 1) compute an estimate for IC regret in an auction, 2) provide a measure of certainty around the estimate of IC regret, and 3) minimize the time it takes to arrive at an accurate estimate.

We approach this problem using tools from the combinatorial (semi-)bandit literature: to measure the IC regret in an auction system, a bidder may enter different bids in different auctions[1]. By judiciously choosing these bids, we give bounds on the difference between the measured IC regret, and the true IC regret of the system as a function of the number of auctions the bidder participated in. This provides a trade-off between the accuracy of the IC regret measurement, and the the amount of time (and thus money) spend on obtaining the insight.

We consider the problem from two perspectives. The first, which we call the "Advertiser Problem," consists of determining the IC regret in the system for some *known* true value v. This is the problem the advertiser faces when they are participating in an online ad auction. In Section 3 we give an algorithm to select bids that determine IC regret with an error that shrinks as $O\Big(\frac{|B|}{T}\cdot\Big(\frac{\ln T}{n}+\sqrt{\frac{\ln T}{n}}\Big)\Big)$, where B is the (finite) bid space, n is the number of auctions at each time step and T is the total number of time steps. The second problem, which we call the "DSP Problem" consists of determining IC regret for the worst-case valuation v, i.e. the value v that yields the *highest* IC regret. This is the problem that a DSP faces if they want to ensure that IC regret is low for *all* advertisers that may use their services. In Section 4 we give an algorithm to select bids and (hypothetical) values that determine worst-case IC regret over all possible values where the error term shrinks as $O\Big(\frac{|B|}{T}\cdot\Big(\frac{|B|\ln T}{n}+\sqrt{\frac{|B|\ln T}{n}}\Big)\Big)$ as the number of time steps T grows.

1.1 Related Work

Structurally, designing incentive compatible (IC) mechanisms is well-understood: if the corresponding optimization problem can be solved exactly then the VCG mechanism [9, 13, 19] is IC. In cases where computing the optimal allocation is prohibitively expensive, and some approximation algorithm is used, VCG is no longer incentive compatible [17]. At the same time, the structure of IC mechanisms beyond optimal allocation is known, and tied to some monotonicity property of the allocation function for single-parameter settings [16] and more generally [18]. A weakness in these approaches is that determining if some given mechanism M is IC, one needs to know both the allocation *function* and the pricing *function*. Without access to the code that computes the allocation and pricing, the best an advertiser (or a DSP who depends on a third party auction that's run by a publisher) can hope to have access to is samples from the allocation and pricing functions, not the functions themselves.

Lahaie et al. [15] are the first to study the problem of testing incentive compatibility assuming only black box access. They design an A/B experiment to determine whether an auction is incentive compatible in both single-shot and dynamic settings (the latter concerns cases where an advertiser's bid is used to set their reserve price in later auctions). The work of Lahaie et al. [15] provides a valuable tool for validation of incentive compatibility, but leaves open the question of how to design an experiment that minimizes the time required to get results with high confidence. The present paper complements [15] by giving an algorithm to select alternative bids that minimize the error in the estimate for IC regret.

Our work is also related with *No regret learning in Game Theory and Mechanism Design* [6]. However, in Game Theory and Mechanism Design, most of the existing literature focuses on maximizing revenue for the auctioneer without knowing a priori the valuations of the bidders, e.g. [1, 2, 4], as well as optimizing the bidding strategy from the bidder side, e.g. [3, 11, 20]. To the best of our knowledge, this is the first paper on testing IC through online learning approach.

1.2 Contributions

There are three main contributions of this paper.

(1) We build an online learning model to measure IC regret with only black-box access to the auction mechanism. We present algorithms that find IC regret with an error that shrinks as $O\Big(\frac{|B|}{T}\cdot\Big(\frac{\ln T}{n}+\sqrt{\frac{\ln T}{n}}\Big)\Big)$ for a known valuation v, and an error that shrinks as $O\Big(\frac{|B|}{T}\cdot\Big(\frac{|B|\ln T}{n}+\sqrt{\frac{|B|\ln T}{n}}\Big)\Big)$ for all possible valuations v. In the above, B is the (finite) bid space, n is the number of auctions at each time step and T is the total number of time steps. We also extend our Regret-UCB algorithm to handle stronger IC regret estimation and shrink the error as $O\left(\frac{|B|^2(\ln T)^{2/3}}{n^{2/3}T}\right)$.

(2) We present a combinatorial semi-bandit algorithm for the setting where the "observed reward" is the max over several arms selected at each time while the benchmark is the standard optimal arm with highest expected reward. To understand this mismatch between the benchmark and observed reward, we analyze the *Pseudo-Regret* of the algorithm and uncover its trade-off with the number of arms selected at each time. This analysis may be of independent interest to the combinatorial (Semi-)Bandits literature.

(3) Simulations suggest that, (1) there is a trade-off between *Pseudo-Regret* and the number of blocks to partition auctions, i.e. we would like to choose the number of blocks to be neither very small nor very large, (2) the *Pseudo-Regret* (near linearly) decays when the number of auctions accessed at each time grows, and (3) our designed algorithm performs better than naive Random-Bids algorithm and ϵ-Greedy algorithm.

[1]In this paper, we consider the model that a bidder participates in many auctions and partitions them into several blocks. For every auction in each block, the bidder enters a bid and only observes the allocation and payment of the bid he enters from each block. For more formal definition, see Section 2.2.

2 MODEL AND PRELIMINARIES

In this section, we formally define the model we considered in this paper and present some important preliminaries. For simplifying description, we introduce "test bidder" to model both the advertiser and DSP.

2.1 Auctions for a Test Bidder

Consider a test bidder i who is eligible for n auctions every day. In each auction, they submit a bid b_i, the auction is run and the outcome is determined by the allocation rule $g_i : b_i \to [0, 1]$ and the payment rule $p_i : b \to \mathbb{R}_{\geq 0}$ (where g_i and p_i are conditioned on the competition in the auction). We assume a single-parameter quasilinear utility model $u_i(v_i, b_i) = g_i(b_i) \cdot v_i - p_i(b_i)$ for some true value $v_i \in \mathbb{R}_{\geq 0}$. Let U be a bound on the utility of a buyer, i.e. the utility function is bounded by $[-U, U]$.

Since there is a large number of auctions and other bidders in the ecosystem, we model the randomness in the system using a stationarity assumption where the conditional allocation and pricing functions are drawn from the same underlying distribution for each auction k:

Assumption 2.1 (Stochastic Assumption). *For each auction k, the allocation rule g_i and payment rule p_i of the test bidder i are drawn from an unknown distribution $\mathcal{F}$, i.e. $(g_i, p_i)_k \overset{i.i.d.}{\sim} \mathcal{F}$.*

2.2 Online Learning Model

At every time step $t \in [T]$, the test bidder participates n auctions and randomly partitions them into $m + 1$ blocks of equal size.[2] For $j \in [m + 1]$, let $\mathcal{A}_j$ be the set of auctions in block j.[3] For every auction in each block j, the test bidder submits a bid $b_t^j \in B$, where B is a finite set of bids.[4]

At the end of time t, in each block $j \in [m]$, the test bidder observes the average allocation probability $\widetilde{g}_t^j(b_t^j)$ and average payment $\widetilde{p}_t^j(b_t^j)$ over all auctions in block j. Let $\vec{b}_t = \{b_t^1, b_t^2, \cdots, b_t^m\}$ be the bids of the test bidder at time t. For block $m + 1$, the test bidder bids their true value v and observes the average allocation $\widetilde{g}_t^{m+1}(v)$ and payment $\widetilde{p}_t^{m+1}(v)$. Without loss of generality, we also assume $v \in B$. The average utility of the test bidder in each block $j \in [m + 1]$ is $\widetilde{u}_t^j(v, b_t^j) \equiv \widetilde{g}_t^j(b_t^j) \cdot v - \widetilde{p}_t^j(b_t^j)$.

With the setting defined, we can instantiate an empirical version of the IC regret from Equation (1) for a given true value v and bids $\vec{b}_t$:

Definition 2.2 (empirical IC regret).

$$\widetilde{rgt}_t(v, \vec{b}_t) = \max_{j \in [m]} \left\{ \widetilde{u}_t^j(v, b_t^j) - \widetilde{u}_t^{m+1}(v, v) \right\}. \tag{2}$$

For notational simplicity, we restate $g^* \triangleq \mathbb{E}[\widetilde{g}_t^j]$ and $p^* \triangleq \mathbb{E}[\widetilde{p}_t^j]$ for each block j at time t.[5] Similarly, define $u^*(v, b) \triangleq \mathbb{E}[\widetilde{u}_t^j(v, b)] \equiv g^*(b) \cdot v - p^*(b)$. Given these definitions, *IC-regret* for a particular true value v and bid b is:

Definition 2.3 (IC-regret). $rgt(v, b) = u^*(v, b) - u^*(v, v)$.

Thus far, in Definitions 2.2 and 2.3 we've considered the regret that a buyer has for bidding their true value v compared to a particular alternative bid b. The quantity that we're really interested in (cf. Equation (1)) is the worst-case IC regret with respect to different possible bids. For the Advertiser Problem (which we treat in Section 3) this is with respect to some known true value v, whereas for the DSP Problem (which we treat in Section 4) we consider the worst-case IC regret over all possible true values v. To summarize, the learning task of the test bidder is to design an efficient learning algorithm to generate v_t and b_t in order to minimize the following *Pseudo-Regret*, the difference between cumulative empirical regret and benchmark.[6]

Definition 2.4 (Pseudo-Regret [5]).

$$\mathbb{E}[R(T)] = \max_{v, b \in B} \sum_{t=1}^{T} rgt(v, b) - \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{rgt}_t(v_t, \vec{b}_t)\right] \tag{3}$$

Given the above *Pseudo-Regret* definition, we can define the error of determining IC regret, which is to measure the distance between the optimal IC regret and the average empirical IC regret over time.

Definition 2.5 (IC Regret Error). $\mathcal{E}(T) = \frac{\mathbb{E}[R(T)]}{T}$

2.3 More Related work

The online learning model in our work is also related to Combinatorial (Semi)-bandits literature [8, 12, 14]. Chen et al. [8] first proposed the general framework and Combinatorial Multi-Armed Bandit problem, and our model lies in this general framework (i.e. the bids are generated from a super arm at each time step). However, in our model the test bidder can observe feedback from multiple blocks at each time step, which is similar to the combinatorial semi-bandits feedback model. Gai et al. [12] analyze the combinatorial semi-bandits problem with linear rewards, and Kveton et al. [14] provides the tight regret bound for the same setting. The Regret-UCB algorithm in our paper is similar and inspired by the algorithm in [12, 14]. However, in our model, the reward function is not linear and has a "max", which needs more work to address it. Chen et al. [7] firstly consider the general reward function in Combinatorial Bandits problem and can handle "max" in reward function, however, there is a mismatch between the benchmark and observed reward in our model (see more in "Contribution").

3 THE ADVERTISER PROBLEM

In this section, we focus on the special case of measuring the IC regret for a known true value v. The learning problem in this setting is to select, for each timestep t, a bid profile $\vec{b}_t$ of m bids to be used in the m auction blocks at that time. The bids $\vec{b}_t$ should be selected to minimize the *Pseudo-Regret* defined in Equation 3.

We propose the Regret-UCB algorithm, given in Algorithm 1, which is inspired by the CombUCB1 algorithm [12, 14]. Regret-UCB works as follows: at each timestep t, the algorithm performs three operations. First, it computes the *upper confidence bound* (UCB) on

[2] The number of blocks m is a variable that will be fixed later. The choice of which m to use will trade off the number of different bids we get information on and the accuracy of the outputs for each bid. Throughout the paper, we assume $m < |B|$.

[3] We let $[n] = \{1, 2, 3, \ldots, n\}$ be the set of positive integers up to and including n.

[4] While in theory the bid space is continuous, in practice outcomes for different bids are relatively smooth, see e.g. the plots in [15], so discretizing the bid space is a reasonable simplification.

[5] Throughout this paper, the expectation $\mathbb{E}$ is over all the randomness (e.g., randomization over payment and allocation distribution and randomization over algorithm).

[6] Where in the case of the Advertiser Problem, we define Pseudo-Regret with respect to a given value v analogously.

the expected utility of each bid $b \in B$ given v,

$$\mathrm{UCB}_t^{\mathrm{u}}(v,b) = \widehat{g}_{t-1}(b) \cdot v - \widehat{p}_{t-1}(b) + 2U\sqrt{\frac{2(m+1)\ln t}{n_{t-1}(b) \cdot n}}, \qquad (4)$$

where $\widehat{g}_s(b)$ is the average allocation probability of bid b up to time s, $n_s(b)$ is the number of times that bid b has been submitted in s steps and U is the bound of utility (i.e. utility is bounded by $[-U, U]$). Second, the algorithm chooses the m bids that correspond to the largest $\mathrm{UCB}_t^{\mathrm{u}}$ values; call this bid vector $\vec{b}_t$. The algorithm uses the m bids $\vec{b}_t$ in the first m blocks of auctions, and in the final block of auctions, it uses the bidder's true value v. Finally, for each of the blocks, the algorithm observes the average allocation probability and payment, and it uses that to update the estimates of allocation $\widehat{g}$ and payment $\widehat{p}$.

Algorithm 1 Regret-UCB Algorithm for a known valuation v.

Input: A finite set of bids B, parameter m, n. $\forall b \in B, n_0(b) = 1$
Initialize: Run $\textsc{Init}(B, m, n)$ algorithm to get $\widehat{g}_0$ and $\widehat{p}_0$.
for $t = 1, \cdots, T$ **do**
  Update UCB^u terms (Eq. (4)) of each bid b.
  Generate a sequence of different m bids $\vec{b}_t \in B^m$ to maximize

$$\sum_{b \in \vec{b}_t} \mathrm{UCB}_t^{\mathrm{u}}(v,b) \qquad (5)$$

  for $j = 1, \cdots, m$ **do**
    $n_t(b_t^j) \leftarrow n_{t-1}(b_t^j) + 1$
    Observe $\tilde{g}_t^j(b_t^j)$ and $\tilde{p}_t^j(b_t^j)$

$$\widehat{g}_t(b_t^j) \leftarrow \left[\widehat{g}_{t-1}(b_t^j) \cdot n_{t-1}(b_t^j) + \tilde{g}_t^j(b_t^j)\right] / n_t(b_t^j)$$

$$\widehat{p}_t(b_t^j) \leftarrow \left[\widehat{p}_{t-1}(b_t^j) \cdot n_{t-1}(b_t^j) + \tilde{p}_t^j(b_t^j)\right] / n_t(b_t^j)$$

  for $b \notin \vec{b}_t$ **do**
    $n_t(b) \leftarrow n_{t-1}(b), \widehat{g}_t(b) \leftarrow \widehat{g}_{t-1}(b), \widehat{p}_t(b) \leftarrow \widehat{p}_{t-1}(b)$

Algorithm 2 Init algorithm to get $\widehat{g}_0$ and $\widehat{p}_0$.

Input: A finite set of bids B, parameter m and n
for b in B **do**
  Randomly participate in $\frac{n}{m+1}$ auctions and submit bid b for each auction.
  Observe the average allocation $\widetilde{g}(b)$ and payment $\widetilde{p}(b)$.
  $\widehat{g}_0(b) \leftarrow \widetilde{g}(b), \widehat{p}_0(b) \leftarrow \widetilde{p}(b)$.

Given a fixed valuation v, define b^* to be the best-response bid:

$$b^* \triangleq \arg\max_{b \in B} rgt(v,b) = \arg\max_{b \in B} u^*(v,b).$$

and denote $\Delta(b) \triangleq u^*(v,b^*) - u^*(v,b)$. The following theorem bounds the worst case *Pseudo-Regret* of the Regret-UCB algorithm for known valuation v.

Theorem 3.1. *Regret-UCB achieves pseudo-regret at most*

$$\sum_{b \in B: u^*(v,b) < u^*(v,b^*)} \frac{32(m+1)U^2 \ln T}{n\Delta(b)} + \frac{\pi^2}{3} \cdot \frac{\Delta(b)}{m}$$

To prove the theorem, we rely on the following lemma, which is widely used in stochastic bandit literature. We refer the reader to check [5] for more material.

Lemma 3.2 ([5]). *Fix a valuation v and an iteration $t-1$ (where $t \geq 2$). If a bid $b \neq b^*$ (i.e. $u^*(v,b) < u^*(v,b^*)$) has been observed $n_{t-1}(b) \geq \frac{8(m+1)(2U)^2 \ln t}{n\Delta(b)^2}$ times, then with probability of at least $1 - \frac{2}{t^2}$, $\mathrm{UCB}_t^u(v,b) \leq \mathrm{UCB}_t^u(v,b^*)$.*

3.1 Discussion

The Regret-UCB algorithm can also be used to implement a low-regret bidding agent: Consider an advertiser who knows the valuation v and wants to maximize the expected utility $u^*(v,b)$ by seeking for a best response bid b. Indeed, the advertiser can adopt the exact same algorithm – Regret-UCB to maximize the utility. The analysis for regret bound in Theorem 3.1 also works when we change the reward function from IC regret to utility.

The term m appears in both terms in Theorem 3.1, hence we can pick it to minimize the asymptotic *Pseudo-Regret*:

Corollary 3.3. *Let $\overline{\Delta} \triangleq \max_{b \in B} \Delta(b)$ and $\underline{\Delta} \triangleq \min_{b \in B} \Delta(b)$. Choosing $m = \frac{\pi}{4U}\sqrt{\frac{n\overline{\Delta}\underline{\Delta}}{6\ln T}}$, the error of determining IC regret achieved by Regret-UCB algorithm for known valuation setting is upper bounded by*

$$\mathcal{E}(T) \leq O\left(\frac{|B|}{T} \cdot \left(\frac{\ln T}{n} + \sqrt{\frac{\ln T}{n}}\right)\right).$$

The big-Oh notation assumes $\overline{\Delta}$ and $\underline{\Delta}$ to be constants.

4 THE DSP PROBLEM

In this section, we consider the problem of determining the worst-case IC regret over all possible valuations v. Let v^* and b^* be the value and bid combination that yields the highest *IC regret*,[7] i.e. $(v^*, b^*) = \arg\max_{v, b \in B} rgt(v,b)$, We modify our previous Regret-UCB algorithm in this setting and show the pseudocode in Algorithm 3.

At each time t, the algorithm first computes the UCBs on the expected *IC regret* of each valuation and bid pair (v,b).

$$\mathrm{UCB}_t^{\mathrm{rgt}}(v,b) = \widehat{rgt}_t(v,b) + 4U\sqrt{\frac{3(m+1)\ln t}{n \cdot \left(n_{t-1}(v) \wedge n_{t-1}(b)\right)}} \qquad (6)$$

where $\widehat{rgt}_t(v,b) = \widehat{g}_t(b) \cdot v - \widehat{p}_t(b) - \left(\widehat{g}_t(v) \cdot v - \widehat{p}_t(v)\right)$, "$\wedge$" is the min function, and the other notations $\widehat{g}_t$, $\widehat{p}_t$, $n_{t-1}(b)$, $n_{t-1}(v)$, and U are identical to Section 3. Then the algorithm selects (v_t, b_t^1) to maximize the $\mathrm{UCB}_t^{\mathrm{rgt}}$ term. Given the valuation v_t, the algorithm chooses other $m-1$ bids to achieve the $(m-1)$–largest $\mathrm{UCB}_t^{\mathrm{u}}(v_t, \cdot)$ terms defined in Equation (4). The rest of update steps in the algorithm are exactly the same as in Algorithm 1.

We denote $\Delta(v,b) \triangleq rgt(v^*,b^*) - rgt(v,b)$ and start with the following lemma that gives the concentration property of the $\mathrm{UCB}^{\mathrm{rgt}}$ terms.

[7]We assume, without loss of generality, that this optimal combination is unique throughout the paper.
[8]We choose the remaining $m-1$ bids be different with each other and b_t^1.

Algorithm 3 Regret-UCB Algorithm for unknown valuation.

Input: A finite set of bids B, parameter m, n. $\forall b \in B, n_0(b) = 1$
Initialize: Run $\textsc{Init}(B, m, n)$ algorithm to get $\widehat{g}_0$ and $\widehat{p}_0$.
for $t = 1, \cdots, T$ **do**
  Update UCB^{rgt} terms (Eq. (6)) of every (v, b) pair
  Choose $(v_t, b_t^1) \in B \times B$ to maximize $\text{UCB}_t^{\text{rgt}}(v, b)$
  if $m \geq 2$ **then**
    Choose remaining $m-1$ bids $\{b_t^2, \cdots, b_t^m\}$ to maximize[8]

$$\sum_{b \in \vec{b}_t \setminus b_t^1} \text{UCB}^{\text{u}}(v_t, b)$$

  for $b \in \vec{b}_t \cup v_t$ **do**
    $n_t(b) \leftarrow n_{t-1}(b) + 1$
    Observe $\tilde{g}_t(b)$ and $\tilde{p}_t(b)$ (omit subscript j for each block),

$$\widehat{g}_t(b) \leftarrow \left[\widehat{g}_{t-1}(b) \cdot n_{t-1}(b) + \widetilde{g}_t^j(b)\right] \Big/ n_t(b)$$

$$\widehat{p}_t(b) \leftarrow \left[\widehat{p}_{t-1}(b) \cdot n_{t-1}(b) + \widetilde{p}_t^j(b)\right] \Big/ n_t(b)$$

  for $b \notin \vec{b}_t$ **do**
    $n_t(b) \leftarrow n_{t-1}(b)$

Lemma 4.1. *At iteration $t-1, (t \geq 2)$, for a (value, bid) pair $(v, b) \neq (v^*, b^*)$, where v and b are both observed at least $\frac{48(m+1)(2U)^2 \ln t}{n\Delta(v,b)^2}$ times, then with probability at least $1 - \frac{4}{t^2}$, $\text{UCB}_t^{rgt}(v, b) \leq \text{UCB}_t^{rgt}(v^*, b^*)$.*

Utilizing Lemma 4.1, we show the worst case *Pseudo-Regret* bound for IC-testing in non-fixed valuation setting in Theorem 4.2.

Theorem 4.2. *Regret-UCB algorithm for unknown valuation setting (DSP problem) achieves pseudo-regret at most*

$$\sum_{\substack{v, b \in B, \\ (v,b) \neq (v^*, b^*)}} \frac{384(m+1)U^2 \ln T}{n\Delta(v, b)} + \frac{2\pi^2 \Delta(v, b)}{3} \cdot \left(\mathbb{1}\left\{v \neq v^*\right\} + \frac{\mathbb{1}\left\{v = v^*\right\}}{m} \right)$$

Following the same argument in Section 3, if we choose m appropriately, we will get the following asymptotic bound.

Corollary 4.3. *Let $\overline{\Delta} \triangleq \max_{v, b \in B} \Delta(v, b)$ and $\underline{\Delta} \triangleq \min_{v, b \in B} \Delta(v, b)$. Choosing $m = \frac{\pi}{24U}\sqrt{\frac{n\overline{\Delta}\underline{\Delta}}{|B| \ln T}}$, the asymptotic error of determining IC regret achieved by Regret-UCB algorithm is upper bounded by*

$$\mathcal{E}(T) \leq O\left(\frac{|B|}{T} \cdot \left(\sqrt{\frac{|B| \ln T}{n}} + \frac{|B| \ln T}{n}\right)\right).$$

The big-Oh notation assumes $\overline{\Delta}$ and $\underline{\Delta}$ to be constants.

Note in the DSP problem, since we consider determining the worst-case IC regret over all possible valuations v, our asymptotic error bound quadratically grows as $|B|$, while for the advertiser problem the error linearly grows as $|B|$ since the benchmark we consider there is weaker than it in the DSP problem.

5 EXTENSION: STRONGER IC REGRET

In Section 4, we assume the test bidder (DSP) only chooses one valuation v_t at every time step t to empirically estimate IC regret. However, in practice, for each (value, bid) pair, the test bidder can switch the roles of bid and value to better estimate the IC regret.

Since there is no conceptual difference between bid and value in this setting, at time step t, the test bidder also submits a "bid" b_t^{m+1} for every auction in block $m+1$. We still denote the bids submitted at time t be $\vec{b}_t = \left\{b_t^1, b_t^2, \cdots, b_t^{m+1}\right\}$. The only difference between this case and our standard model is from the definition of *empirical IC regret*, we define the modified *empirical IC regret* in this setting as below,

Definition 5.1 (modified empirical IC regret).

$$\widetilde{rgt}_t(\vec{b}_t) = \max_{i, j \in [m+1], i \neq j} \left\{\widetilde{u}_t^j(b_t^i, b_t^j) - \widetilde{u}_t^i(b_t^i, b_t^i)\right\}$$

The learning task in this setting is to design an efficient algorithm to generate $m+1$ bids at every time step to minimize *Pseudo-Regret*,

$$\mathbb{E}\left[R(T)\right] \triangleq \max_{v, b \in B} \sum_{t=1}^{T} rgt(v, b) - \mathbb{E}\left[\sum_{t=1}^{T} \widetilde{rgt}_t(\vec{b}_t)\right]$$

We extend our Regret-UCB algorithm for unknown valuation (Algorithm 3) for this setting.

At each time step t, we update the UCB^{rgt} terms defined in Equation (6) for every (b_1, b_2) bids pair. Given these UCB^{rgt} terms, we run Generate-Bids algorithm proposed in Algorithm 4 to generate $m+1$ bids and update $\hat{p}_t(b)$ and $\hat{g}_t(b)$ for any chosen b at time t.

Algorithm 4 Generate-Bids Algorithm for generating bids given UCB^{rgt} terms.

Input: A finite set of bids B, UCB^{rgt} terms for every (b_1, b_2) bids pair, where $b_1, b_2 \in B$. Initialize a set $B_0 = \{\}$.
while $|B_0| < m+1$ **do**
  Choose the largest $\text{UCB}^{\text{rgt}}(b_1, b_2)$ such at $b_1 \notin B_0$ or $b_2 \notin B_0$.
  Update $B_0 \leftarrow B_0 \cup \{b_1, b_2\}$. [9]

We call the above algorithm for stronger IC regret estimation Regret-UCB*, of which the *Pseudo-Regret* is shown in Theorem 5.2.

Theorem 5.2. *Regret-UCB* algorithm for DSP problem using stronger IC regret estimation achieves pseudo-regret at most*

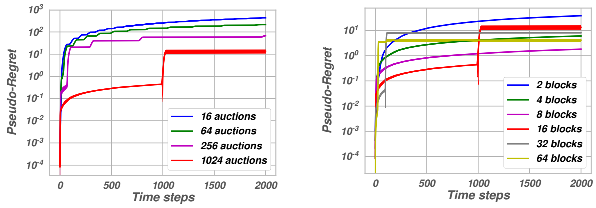
$$\sum_{\substack{b_1, b_2 \in B, \\ (b_1, b_2) \neq (v^*, b^*)}} \frac{192(m+1)U^2 \ln T}{n\Delta(b_1, b_2)} + \frac{2\pi^2 \Delta(b_1, b_2)}{3(m+1)^2}$$

Note the *Pseudo-Regret* bound achieved by Regret-UCB* algorithm is always better than it achieved by Algorithm 3 for any m. Then we can always get a better asymptotic error bound by Regret-UCB* algorithm for the DSP problem.

Corollary 5.3. *Choosing $m = O\left(\frac{n}{\ln T}\right)^{1/3}$ the asymptotic error of determining IC regret achieved by Regret-UCB algorithm is upper bounded by*[10]

$$\mathcal{E}(T) \leq O\left(\frac{|B|^2 (\ln T)^{2/3}}{n^{2/3} T}\right)$$

[9] If the two bids chosen in the last round are both not in B_0, we randomly choose one to add it in B_0.
[10] This asymptotic error bound can be achieved if the optimal m can be chosen to satisfy the constraint that $m < |B|$.



(a) (b)

Figure 1: *Pseudo-Regret* of Regret-UCB algorithm for known valuation $v = 9.5$ (Algorithm 1).

6 SIMULATIONS

In this section, we run some simulations with Generalized Second Price (GSP) auctions to validate our theoretical results. Every experiment is repeated for 10 times and the dash area represents the 95% confidence intervals of *Pseudo-Regret* at each time step.

6.1 Settings

At each time t, the test bidder randomly participates n GSP auctions. Each auction has 5 slots and 20 bidders, each bidder's bid is i.i.d generated from $U[0, 10]$. The *click-through-rate* sequence of each auction is generated from descending ordered $Beta(2, 5)$ distribution[11]. We consider a finite bids space $B \triangleq \{0.01, 0.02, \cdots, 10\}$.

6.2 The advertiser problem

We test the performance of Regret-UCB algorithm for known valuation $v = 9.5$. First, we fix the number of blocks be 16 (i.e. $m = 15$) and plot the *Pseudo-Regret* achieved by Regret-UCB with different $n = 16, 64, 256, 1024$ (Figure 1(a)). We observe that the *Pseudo-Regret* decays linearly with n which is consistent with our analysis. Second, we fix $n = 1024$ and test the performance of the algorithm with different number of blocks, such as $m = 1, 3, 7, 15, 31, 63$ (Figure 1(a)). We find $m = 7$ (8 blocks) achieves the lowest *Pseudo-Regret*. When m is too large, the *Pseudo-Regret* curve incurs some shocks and suffers high variance because of the noisy observed information at each time.

6.3 The DSP problem

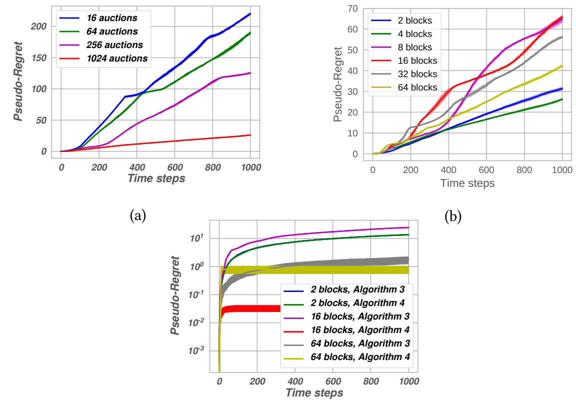
We test the performance of Regret-UCB for unknown valuation case (i.e., the DSP problem). Similarly, we show the *Pseudo-Regret* curve for different n given $m = 15$ and different m given $n = 1024$ in Figure 2. Figure 2(a) validates that *Pseudo-Regret* decays when n grows and we observe $m = 3$ (4 blocks) gives the best *Pseudo-Regret*. This corresponds to our theory that the optimal m in the DSP problem should be smaller than the optimal m in the advertiser problem for the same auction setting when $|B|$ is large (see Corollary (3.3) and (4.3)).

6.4 Efficiency of Regret-UCB algorithm

We first introduce two standard baselines, Random-Bids algorithm and ϵ-Greedy algorithm.

Random-Bids. At each time step t, the advertiser uniformly randomly choose m bids b_t, while the DSP uniformly randomly choose m bids b_t and a valuation v_t from B.

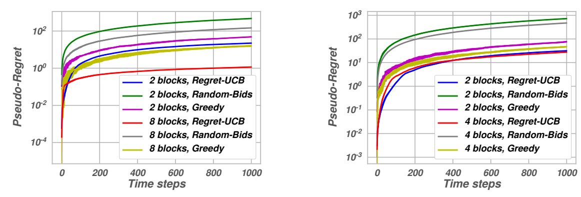
[11] *Click-through-rate* of each slot is first i.i.d generated from $Beta(2, 5)$ and then arranged by descending order.



(a) (b)

(c)

Figure 2: *Pseudo-Regret* plot of Regret-UCB algorithm for unknown valuation. (a): Algorithm 3 for different n, (b): Algorithm 3 for different m and (c): a semi-logarithmic *Pseudo-Regret* to compare Algorithm 3 and Regret-UCB*.



(a) (b)

Figure 3: *Pseudo-Regret* with Random-Bids, ϵ-Greedy and Regret-UCB for the advertiser problem (a) and DSP problem (b).

ϵ-Greedy. For the advertiser problem with known valuation v, at each time step, the advertiser uniformly randomly chooses m bids from B with probability ϵ, otherwise, chooses the m bids that correspond to the largest average utility $\hat{u}(v, \cdot)$ terms.[12] For the DSP problem, at each time step t, the DSP uniformly randomly choose m bids b_t and a valuation v_t with probability ϵ. Otherwise, the DSP chooses the (v_t, b_t^1) pair corresponds to the largest $\widehat{rgt}$ terms and the rest $m - 1$ bids associated with the largest $\hat{u}(v_t, \cdot)$ terms.

We compare the performance of our Regret-UCB algorithm with the above two baselines for the advertiser problem and the DSP problem. For the both settings, Regret-UCB algorithm performs better than two baselines for different number of blocks.

6.5 Regret-UCB for stronger IC regret

Moreover, we compare the performance of Algorithm 3 and Regret-UCB* algorithm for different m given $n = 1024$ in Figure 2(c) to validate the theory that allowing switching value and bid leads to lower Pseudo-Regret.[13] In addition, we also observe the tradeoff between Pseudo-Regret and m of Regret-UCB* algorithm. Among $m = 1, 15, 63$, the optimal m is 15 (16 blocks) for Regret-UCB*

[12] In the experiments, we fix $\epsilon = 0.1$

[13] Based on our observation in the experiments, sometimes switching value and bid may "over-estimate" the IC regret, i.e. the empirical estimation of IC regret is sometimes larger than worst case IC regret.

algorithm. If we choose m too large, like $m = 63$, Pseudo-Regret is worse than the $m = 1$ case and incurs high variance.

REFERENCES

[1] Kareem Amin, Rachel Cummings, Lili Dworkin, Michael Kearns, and Aaron Roth. 2015. Online Learning and Profit Maximization from Revealed Preferences. In *AAAI*. 770–776.

[2] Kareem Amin, Afshin Rostamizadeh, and Umar Syed. 2014. Repeated contextual auctions with strategic buyers. In *Advances in Neural Information Processing Systems*. 622–630.

[3] Santiago Balseiro and Yonatan Gur. 2017. Learning in Repeated Auctions with Budgets: Regret Minimization and Equilibrium. (2017).

[4] Avrim Blum, Vijay Kumar, Atri Rudra, and Felix Wu. 2004. Online learning in online auctions. *Theoretical Computer Science* 324, 2-3 (2004), 137–146.

[5] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5, 1 (2012), 1–122.

[6] Shuchi Chawla, Jason D. Hartline, and Denis Nekipelov. 2014. Mechanism design for data science. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*. 711–712.

[7] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. 2016. Combinatorial multi-armed bandit with general reward functions. In *Advances in Neural Information Processing Systems*. 1659–1667.

[8] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*. 151–159.

[9] Edward H Clarke. 1971. Multipart pricing of public goods. *Public choice* 11, 1 (1971), 17–33.

[10] Vincent Conitzer, Christian Kroer, Eric Sodomka, and Nicolás E. Stier Moses. 2017. Multiplicative Pacing Equilibria in Auction Markets. *CoRR* abs/1706.07151 (2017). arXiv:1706.07151 http://arxiv.org/abs/1706.07151

[11] Zhe Feng, Chara Podimata, and Vasilis Syrgkanis. 2018. Learning to Bid Without Knowing Your Value. In *Proceedings of the 2018 ACM Conference on Economics and Computation (EC '18)*. ACM, New York, NY, USA, 505–522.

[12] Y. Gai, B. Krishnamachari, and R. Jain. 2012. Combinatorial Network Optimization With Unknown Variables: Multi-Armed Bandits With Linear Rewards and Individual Observations. *IEEE/ACM Transactions on Networking* 20, 5 (Oct 2012), 1466–1478.

[13] Theodore Groves. 1973. Incentives in teams. *Econometrica: Journal of the Econometric Society* (1973), 617–631.

[14] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*. 535–543.

[15] Sébastien Lahaie, Andrés Munoz Medina, Balasubramanian Sivan, and Sergei Vassilvitskii. 2018. Testing Incentive Compatibility in Display Ad Auctions. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1419–1428.

[16] Roger B Myerson. 1981. Optimal auction design. *Mathematics of operations research* 6, 1 (1981), 58–73.

[17] Noam Nisan and Amir Ronen. 2007. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research* 29 (2007), 19–47.

[18] Jean-Charles Rochet. 1987. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of mathematical Economics* 16, 2 (1987), 191–200.

[19] William Vickrey. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance* 16, 1 (1961), 8–37.

[20] Jonathan Weed, Vianney Perchet, and Philippe Rigollet. 2016. Online learning in repeated auctions. In *Conference on Learning Theory*. 1562–1583.