Optimized Cost per Mille in Feeds Advertising

Pingzhong Tang
Tsinghua University
Beijing, China
kenshinping@gmail.com

Xun Wang
Tsinghua University
Beijing, China
wxhelloworld@outlook.com

Zihe Wang
ITCS, SUFE
Shanghai, China
wang.zihe@mail.shufe.edu.cn

Yadong Xu
Tsinghua University
Beijing, China
xuyd17@mails.tsinghua.edu.cn

Xiwang Yang
ByteDance
Beijing, China
yangxiwang@bytedance.com

ABSTRACT

Advertising has become a dominant source of revenue generation on the Internet. Billions of advertisement slots are sold via auctions. And there are many pricing methods ,e.g., CPM (cost-per-mille), CPC (cost-per-click), CPA (cost-per-action), OCPM (optimized cost-per-mille) and so on. In this paper, we study the OCPM method (i.e., advertisers bid for conversions while pay per mille) under VCG auction. However, automatically bid in each view to maximize advertisers' conversions while still meet their target cost-per-conversion in feeds is difficult. To deal with these difficulties, we propose a reinforcement learning framework, i.e., RSDRL (ROI-sensitive distributional reinforcement learning). By making full use of the characteristics of auction rules which are missed by other methods, we design a reward function to surrogate conversion events and a bid generation method based on theoretical results. We also provide some theoretical results to guide hyperparameter tuning. Last, we validate RSDRL on a large industrial dataset with millions of auctions. Plenty of experiments (both online and offline) are used to evaluate the performance of our framework and RSDRL yields substantially better results than compared algorithms.

KEYWORDS

VCG; OCPM; reinforcement learning

ACM Reference Format:
Pingzhong Tang, Xun Wang, Zihe Wang, Yadong Xu, and Xiwang Yang. 2020. Optimized Cost per Mille in Feeds Advertising. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Internet advertising has been one of the most important research fields at the interface of AI and economics. It is a trillion-dollar market and still rapidly growing. Billions of advertisement slots are sold via auctions. In industry, the widely used auctions are VCG [5, 12, 21] (e.g., Facebook, Bytedance) and GSP [9] (e.g., Google, Baidu). And it is proved that GSP has a Nash equilibrium whose outcome is equivalent to VCG. The platform provides different pricing methods so advertisers are able to choose any of them according to their commercial purposes. There are three traditional pricing methods in auction, i.e., CPM, CPC and CPA. To be specific, CPM is more suitable for brand promotions and maintaining brand awareness, while CPC and CPA are more suitable for immediate sales growth. Recently, to better meet different commercial purposes, there are many more pricing methods proposed like ECPC (enhanced cost-per-click) in Google[1], OCPM (optimized cost-per-mille) and OCPC (optimized cost per click) in Facebook[2], Alibaba[3] and so on. All these new pricing methods try to optimize conversions compared with traditional methods.

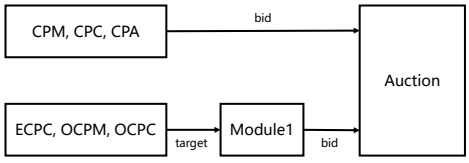

Figure 1: Different pricing methods in advertising system

We use Figure 1 to demonstrate how these pricing methods work in advertising system. Both 'Module 1' and 'Auction' belong to advertising system. The CPM, CPC and CPA advertisers need to manually bid based on their target. While for ECPC, OCPM and OCPC advertisers, they only need to set a target. Although they have to pay per click or per mille, the platform will automatically bid for them to meet their target, i.e., the cost-per-conversion cannot exceed a certain value. Compared with those traditional methods, there are many advantages from the perspectives of both sides.

- As for advertisers, these pricing methods make bid optimization convenient. The platform has to be responsible for their revenue, i.e., conversions, and achieve finer matching of bid and traffic quality of page view (PV) request granularity.
- As for platform, these pricing methods can transfer risks for the uncertainty of conversion to advertisers. In CPA, advertisers have less incentive to provide attractive advertising context because they only have to pay when the conversion occurs. However, in these new pricing methods, e.g., OCPM, platform would automatically decreases the bids for advertisers whose ads have a low probability of conversions in each auction to meet their

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

[1] https://support.google.com/google-ads/
[2] https://www.facebook.com/business/help/494633817315490
[3] https://www.alimama.com/index.htm

cost-per-conversion targets. Thus, these advertisers would win fewer auctions and get fewer conversions unless they provide more attractive advertising context to improve their conversion rate .

Thus, bid optimization is a very important part within Internet advertising from both the advertiser side and the platform side. As for advertisers, they want to use less money while win more auctions; As for platform, there are more objectives, ranging from revenue maximization [3, 16, 20, 24], to meeting advertisers' KPI for a better advertising ecosystem [4]. Some researchers study bidding strategy from the perspective of game theory, e.g., [8, 17–19]. However, they usually rely on knowing each advertisers' value distribution and assume that advertisers follow the Nash equilibrium; Some researchers study bidding strategy from the perspective of traditional optimization methods, e.g., [10] presents a linear programming (LP)-based polynomial time algorithm. However, these methods have difficulty to deal with new advertisers; Some researchers study bidding strategy from the perspective of learning, e.g., [14] learns to strategically bid and fools the platform to set lower reserve; [1] casts this problem as a Markov Decision Process with censored observations.

We study OCPM in feeds from the perspective of learning, via reinforcement learning. Although this has been studied in the original sponsored search auction, there are many related work [23, 26, 27] and all of them have good performance on real data. However, there are still two differences. Firstly, advertising in feeds is quite different:

- The number of advertisement slots in each query is not fixed. What's more, the position of advertisement slots in each query can also be different. Thus, advertising in feeds is a more complicated dynamic environment.
- The allocation structure in feeds differs from that in the original sponsored search auction. In feeds, if the sponsored messages are presented in a coherent context, then the interaction rate (i.e., CTR, CVR) can be different. For example, Nike ads would get more attention when surrounded by sports news.

These call for a new model for bid optimization of OCPM advertisers in feeds, rather than those already used; Secondly, we further improve the performance of reinforcement learning by making full use of the characteristics of auction rules which are missed by these methods. Our contributions can be summarized as follows.

- We propose a new model, i.e., RSDRL (ROI-sensitive distributional reinforcement learning) to deal with this problem and also design a new reward function and a bid generation method based on theoretical results. Besides, we also provide some theoretical results to guide hyperparameter tuning.
- We testify the effectivity of RSDRL on a large industrial dataset with millions of auctions through many experiments.

1.1 Additional related work

We briefly review two fields related to our work.

- **Bid optimization in RTB**: In the RTB (real-time bidding) process, the advertiser receives the bid request of an ad impression with its real-time information and the very first thing to do is to set right bid to maximize key performance indicator (KPI) such as conversion or profit [22]. There is a growing body of research that studies this problem [4, 11, 15, 25, 26].
  In [26], they propose a functional framework to optimize bidding strategy. However, their results rely on an assumption that the auction winning function has a consistent concave shape form; In [28], they propose a bid optimization approach which is polynomial-time and can achieve the overall optimization of advertisers' interests, user experience and platform revenue; [23] provide a deep reinforcement learning algorithm and they sample an auction in every 100 auctions interval as the next state; [27] also propose a deep reinforcement learning framework. They observe that auction sequences of two days share similar transition patterns at a proper aggregation level. So they formulate their model at hour-aggregation level of the auction data.
- **Distributional reinforcement learning**: Distributional RL models the distribution of returns from a state instead of only its expected value. The first distributional RL algorithm, i.e., C51, makes great improvement on the Atari-57 benchmark when compared with previous DQN variants [2]. Subsequently, [13] combined C51 with enhancements such as prioritized experience replay, n-step updates, and the dueling architecture. [7] and [6] improve the results of C51 by using quantile regression to approximate the full quantile function for the state-action return distribution.

2 PRELIMINARIES

In this section, we will formulate the problem of OCPM bid optimization in feeds.

A sequence of news items is presented to a user as he scrolls down the screen of a smartphone. Many ad slots are inserted among them. Those slots are sold via VCG auction and advertisers would bid for them. These advertisers can select different pricing methods for different business purposes [4], e.g., CPC (cost per click), CPM (cost per mille), OCPM (optimized cost per mille) and so on. Here, we only focus on the bid optimization for OCPM advertisers. In OCPM pricing method, advertisers can bid for conversion and actually pay per mille. First, advertiser can set a target cost-per-conversion; Then OCPM pricing method works by automatically bidding for each view and ensures that advertisers only pay when their ads can be seen. And most importantly, the average cost-per-conversion is kept below the target.

Let $\mathcal{I}_i$ be the set of auctions which advertiser i participates in. If $l, l' \in \mathcal{I}_i$ and $l < l'$, then i participates in l before l'. Hence we can assume $\mathcal{I}_i = \{1, 2, \ldots, |\mathcal{I}_i|\}$ without loss of generality. For $l \in \mathcal{I}_i$, we use $pctr^l_{i,j}$ and $pcvr^l_{i,j}$ to denote the predicted click-through rate and predicted conversion rate for i if he is allocated to slot j in auction l. If $j < j'$, then $pctr^l_{i,j} \geq pctr^l_{i,j'}$ and $pcvr^l_{i,j} \geq pcvr^l_{i,j'}$. We use b^l_i to denote the bid for advertiser i in auction l. For OCPM advertiser i, we can mathematically formulate his objective as Equation (1) where $\alpha^l_i \in \{0, 1\}$ is a binary variable to denote the conversion number in auction l, p^l_i denotes his cost in l and v_i denotes the target cost-per-conversion. The only constraint in Equation (1) indicates that the average cost-per-conversion should

[4] As far as we known, it's quite common in industry.

be no greater than v_i.

$$\max \quad \sum_{l=1}^{|\mathcal{I}_i|} \alpha_i^l \tag{1}$$
$$s.t. \quad \frac{\sum_{l=1}^{|\mathcal{I}_i|} p_i^l}{\sum_{l=1}^{|\mathcal{I}_i|} \alpha_i^l} \le v_i$$

The representation of ROI roi_i^l is derived as Equation (2), which denotes the ratio of average cost-per-conversion to v_i for the first l auctions.

$$roi_i^l = \frac{\sum_{l'=1}^{l} p_i^{l'}}{v_i \sum_{l'=1}^{l} \alpha_i^{l'}} \tag{2}$$

2.1 VCG auction

It consists of two functions $M = (\boldsymbol{\sigma}, \boldsymbol{p})$, where the allocation rule is a function $\boldsymbol{\sigma} : R^n \to R^n$, which takes advertisers' bids as input and outputs an n-dimensional vector indicating the slot allocated to each advertiser (i.e., $\sigma_i = j \le m$ denotes advertiser i is allocated to slot j and $\sigma_i = m+1$ denotes advertiser i does not win any slot). The payment rule is a function $\boldsymbol{p} : R^n \to R^n$ that maps advertisers' bids to an n-dimensional non-negative vector specifying the payment of each advertiser (i.e., advertiser i has to pay p_i if he wins slot σ_i). To simplify notations, we also use $\boldsymbol{\sigma}$ and $\boldsymbol{p}$ to denote the allocation and payment, respectively.

In auction l, there are 4 (i.e., CPC, CPM, CPA, OCPM) kinds of advertisers. The allocation $\boldsymbol{\sigma}^l$ is determined by solving Equation (3). The β^l_{i,σ_i^l} is different for different pricing method, that is, $\beta^l_{i,\sigma_i^l} = pctr^l_{i,\sigma_i^l} \cdot pcvr^l_{i,\sigma_i^l}$ if i is an OCPM advertiser or a CPA advertiser; $\beta^l_{i,\sigma_i^l} = pctr^l_{i,\sigma_i^l}$ if i is a CPC advertiser; $\beta^l_{i,\sigma_i^l} = 1$ if i is a CPM advertiser. Thus, $\beta^l_{i,\sigma_i^l} \cdot b_i$ denotes the expected bid based on the corresponding pricing method. It's worth noting that $\beta^l_{i,\sigma_i^l} \ge \beta^l_{i,\bar{\sigma}_i^l}$ holds if $\sigma_i^l \le \bar{\sigma}_i^l$.

$$\boldsymbol{\sigma}^l \in \arg\max_{\bar{\boldsymbol{\sigma}}^l} \sum_{i=1}^{n} \beta^l_{i,\bar{\sigma}_i^l} \cdot b_i^l \tag{3}$$

The payment $\boldsymbol{p}^l$ is determined by Equation (4).

$$p_{i'}^l = \max_{\bar{\sigma}^l} \sum_{i=1, i\neq i'}^{n} \beta^l_{i,\bar{\sigma}_i^l} \cdot b_i^l - \sum_{i=1, i\neq i'}^{n} \beta^l_{i,\sigma_i^l} \cdot b_i^l. \tag{4}$$

In a word, VCG finds an allocation to maximize the expected reported bids (i.e., social welfare) and charges each advertiser for the cost his presence imposes on the other advertisers.

Lemma 2.1. *The objective of CPC, CPM, CPA advertisers in each auction can be mathematically formulate as Equation* (5)*. And they have a dominant bid strategy in VCG auction [5, 12, 21].*

$$u_i^l = \beta^l_{i,\sigma_i^l} b_i^l - p_i^l \tag{5}$$

Lemma 2.1 denotes that for advertisers with other pricing methods, there is a dominant bid strategy for them in VCG auction. Here, a dominant strategy is better than another strategies for one advertiser, no matter how his opponents may bid. *Thus during the optimization, we do not need to worry about that these advertisers would change their bids.*

2.2 Model

Based on the before mentioned OCPM bid optimization problem, we now formulate it into a reinforcement learning model. In auction l,

- **State** s_i^l: For advertiser i, the state is represented as $s_i^l =< v_i, t, roi_i^l, \overrightarrow{auct} >$, where t denotes the current time, and $\overrightarrow{auct}$ is the feature vector related to the auction that we can get from the advertising environment.
- **Action** a_i^l: The bid.
- **Reward** $r_i(s_i^l, a_i^l)$: The income gained according to a specific action a_i^l under state s_i^l.
- **Policy** $\pi(s_i^l)$: Action $\pi(s_i^l)$ should be taken under state s_i^l.
- **Episode** ep: In this paper, we treat one day as an episode.

Finally, our goal is to find a policy $\pi(\cdot)$ which determines the action a_i^l under state s_i^l to maximize the expected accumulated rewards: $\sum_{l=1}^{|\mathcal{I}_i|} \gamma^{l-1} r_i(s_i^l, a_i^l)$, where γ is the discount rate used in a standard RL model.

3 REWARD FUNCTION DESIGN

In this section, we will first discuss the difficulties of designing reward function; Then a specific reward function will be proposed according to some theoretical results.

In our problem, the goal of OCPM advertiser is to maximize his conversions with the cost-per-conversion constraint (Equation (1)). However, *using conversions to design an appropriate reward function has the following difficulties.*

- **Conversion only provides limited information for training since it is an event with low probability**: As for advertiser i in auction l, the probability of conversion equals $pctr^l_{i,\sigma_i^l} \cdot pcvr^l_{i,\sigma_i^l}$. It's not difficult to imagine that different actions (i.e., bids) may lead to the same result (i.e., zero conversion) since it is an event with low probability in industry.
- **The cost-per-conversion constraint**: This constraint is critical in the OCPM pricing method. Without the cost-per-conversion constraint, the bid optimization would be trivial (i.e., bid high enough to win the first slot in each auction). However in Equation (1), we need to balance the conversion number against the corresponding cost, which contributes to the difficulty in reward function design. Using conversion number as reward function cannot provide this information.

3.1 A reward design methodology

It becomes crucial to design a new reward function that is simple enough and can handle difficulties mentioned before. And we handle them as:

- The reward function is designed in terms of advertiser's payment instead of his conversions based on Theorem 3.1.
- We deal with the cost-per-conversion constraint by adding penalty in terms of advertiser's payment. This penalty has theoretical guarantee as we mentioned in Section 6.

Theorem 3.1 implies that higher payment means more expected conversions. *And it provides the design implication that the reward function can be formulated in terms of advertiser's payment.*

Theorem 3.1. *In VCG auction l, for OCPM advertiser i, the following three statements are equivalent.*

$$(1)E[\alpha_i^l] \geq E[\hat{\alpha}_i^l], \quad (2)b_i^l \geq \hat{b}_i^l, \quad (3)p_i^l \geq \hat{p}_i^l$$

where if advertiser i bids b_i (or $\hat{b}_i$), then he would get $E[\alpha_i^l]$ (or $E[\hat{\alpha}_i^l]$) conversion and has to pay p_i^l (or $\hat{p}_i^l$).

Due to limited space, we omit the proof here. Thus, we formulate the reward function as Equation (6) where p_i^l denotes the payment of i when he uses a_i^l to generate bid.

$$r_i(s_i^l, a_i^l) = p_i^l - \max\{\lambda(p_i^l - \beta^l_{i,\sigma_i^l} \cdot v_i), 0\} \tag{6}$$

where λ is a positive constant. Equation (6) can be maximized when p_i^l is equal to $\beta^l_{i,\sigma_i^l} \cdot v_i$. We use β^l_{i,σ_i^l} (i.e., the predicted value of $E[\alpha_i^l]$) instead of α_i^l because α_i^l is 0 with a high probability (i.e., conversion happens rarely). So using α_i^l only provides limited information for training as we mentioned before. Actually, with a carefully selected λ, this reward would be negative in each auction if the cost-per-conversion constraint is broken. There will be more details on the choice of λ in Section 6.

4 BID GENERATION

In this section, firstly, the bid generation method in RSDRL will be introduced; Then we show how the relaxation (i.e., using β^l_{i,σ_i^l} instead of α_i^l in Equation (6)) can violate the cost-per-conversion constraint; Finally, we provide an algorithm to deal with this violation.

Given action a_i^l, the bid takes the form of Equation (7). Instead of generating bid directly, the bid is the product of two parts, the base-bid part (i.e., v_i) and the adjustment part (i.e., a_i^l). *The main reason is that the value of bid is not instructive.* It's not hard to imagine that advertiser i who sells notebooks needs a small bid to win an auction while advertiser $\hat{i}$ who sells cars needs a huge bid to win an auction. What counts is the ratio of bid to cost-per-conversion target (i.e., v_i) in the complicated dynamic environment.

$$b_i^l = v_i \cdot (1 + a_i^l) \tag{7}$$

4.1 Cost-per-conversion constraint violation

Using β^l_{i,σ_i^l} instead of α_i^l could violate the cost-per-conversion constraint . *The reason is that the predicted conversion number can be overvalued.* We mathematically formulate this violation as Equation (8).

$$\begin{cases} \sum_{l=1}^{|\mathcal{I}_i|} p_i^l \leq v_i \sum_{l=1}^{|\mathcal{I}_i|} \beta^l_{i,\sigma_i^l} \\ \sum_{l=1}^{|\mathcal{I}_i|} p_i^l > v_i \sum_{l=1}^{|\mathcal{I}_i|} \alpha_i^l \end{cases} \tag{8}$$

The first equation in Equation (8) implies that the cost per conversion is less than v_i if conversion number equals the predicted value. However, the second equation implies that the cost per conversion is greater than v_i actually.

We also use real data to demonstrate this phenomenon. Let pcvr_diff be Equation (9). *The value of pcvr_diff is less than* 1 *if the predicted conversion number is overvalued.* We illustrate the pcvr_diff of 320363 advertisers in Table 1. There is a gap between predicted conversions and real conversions. The pcvr_diff of 71% advertisers is less than 0.8; The pcvr_diff of 12% advertisers is more than 1.2; And the pcvr_diff of 18% advertisers is between 0.8 and 1.2.

$$\text{pcvr\_diff} = \frac{\sum_{l=1}^{|\mathcal{I}_i|} \alpha_i^l}{\sum_{l=1}^{|\mathcal{I}_i|} \beta^l_{i,\sigma_i^l}} \tag{9}$$

Table 1: The proportion of different pcvr_diff

The pcvr_diff range	Proportion
$[1.2, +\infty)$	12%
$(0.8, 1.2)$	18%
$[0, 0.8]$	71%

There are two reasons for this difference.

- Although CTR (or CVR) predication has been widely studied, it's still a central problem in the computational advertising domain. There are too many influence factors, i.e., user behaviors, ads features and so on. In feeds, this predication can be more complicated because the externalities feed items shown in other positions may impose on the probability that an ad in a particular position receives a click.
- In auction l, for OCPM advertiser i, α_i^l can be regarded as a sample drawn from a binomial distribution where the probability of success equals β^l_{i,σ_i^l}. Even the predication is accurate, there still exists variance.

Of course, we do not provide an algorithm that can make an accurate predication. In what follows, we propose a method to prevent the cost-per-conversion constraint violation and use experiments to verify its effectivity.

4.2 ROI-sensitive agent

Distributional reinforcement learning provides an effective way to learn the state-action return distribution. By re-parameterizing a distribution over the sample space, this gives rise to a large class of risk-sensitive policies. Here, 'risk' refers to the uncertainty over possible outcomes, and risk-sensitive policies are those which depend on more than the expectation of the state-action return distribution.

Here, we build our algorithm based on IQN [6], a well-known distributional reinforcement learning which learns an implicit representation of the return distribution by using quantile values. Let $Q_\tau(s, a)$ be the quantile function at $\tau \sim U([0, 1])$ for the random variable $Q(s, a)$. Let $\rho : [0, 1] \to [0, 1]$ be a distortion risk measure. Then, the expectation of $Q(s, a)$ under $\rho(\cdot)$ is given by Equation (10).

$$Q_\rho(s, a) = \mathbb{E}_{\tau \sim U([0,1])}[Q_{\rho(\tau)}(s, a)] \tag{10}$$

Denote by π_ρ the policy under $\rho(\cdot)$ as Equation (11).

$$\pi_\rho(s) = \arg\max_a Q_\rho(s, a) \tag{11}$$

This provides an algorithmic implication. With the change of roi_i^l (Equation (2)), we can use different $\rho(\cdot)$. For example, if roi_i^l is high, $\rho(\cdot)$ can give more weights to small quantiles of $Q(s,a)$. $\rho(\cdot)$ takes the form of Equation (12).

$$\rho(\tau) = \begin{cases} \tau & \text{if} \quad roi_i^l \le \theta \\ \min\{\tau, \hat{\tau} \sim U([0,1])\} & \text{otherwise} \end{cases} \tag{12}$$

where θ is a pre-defined constant. When roi_i^l is higher than θ, this is a risk-avoiding attitude which emphasizes low returns. Thus, the agent is willing to take a loss relative to the expected return in exchange for certainty. Based on Theorem 3.1 and the reward function (Equation (6)), agent would bid small if roi_i^l is higher than θ. We use this method to model the ROI-sensitive agent in RSDRL framework.

5 ALGORITHM

Now we present our algorithm. It is built based on IQN actually. The process an agent interacting with the auction system within RSDRL can be illustrated in Figure 2. Comparing with the interacting process of traditional reinforcement learning, there are two differences.

- ROI update process: Before the ROI-sensitive agent makes his decision in auction l, roi_i^l will be updated (Equation (2)). Then based on roi_i^l, his attitude to risk will be determined.
- Bid generation process: The action is an adjustment factor and the bid is generated according to Equation (7).

The complete ROI-sensitive distributional reinforcement learning framework is designed for OCPM advertiser i and presented in Algorithm 1.

Algorithm 1 RSDRL

Randomly initialize weights μ for network Q
Randomly initialize weights $\mu' = \mu$ for target network Q'
Initialize replay memory D
Initialize $roi_i^l = 0$

for episode = 1 to K **do**
  for l=1 to $|I_i|$ **do**
    Get ρ based on Equation (12)
    With probability ϵ select a random action a_i^l
    Otherwise get action a_i^l according to Equation (11)
    Bid with $v_i \cdot (1 + a_i^l)$
    Get reward r_i^l
    Observe next state s_i^{l+1}
    Store transition $(s_i^l, s_i^{l+1}, a_i^l, r_i^l)$ in D
    Update roi_i^l
    Sample random mini-batch of transitions from D
    Perform a gradient descent step on IQN loss with respect to the μ
    Every C steps reset $Q' = Q$
  end for
end for

During the inner loop of the algorithm, the ROI-sensitive agent selects and executes actions according to an ϵ-greedy policy based on $Q_\rho(\cdot)$. Then, bids are produced for the OCPM advertiser to compete with other bidders. Based on the allocation rule and payment rule of VCG auction, reward and the next state can be obtained. When the ad is presented to a user, α_i^l can also be obtained. Thus we can update the value of roi_i^l. Finally, the network is updated by performing a gradient descent according to the IQN loss [6] calculated based on a mini-batch of $(s, a, s', r(s,a))$ sampled from experience.

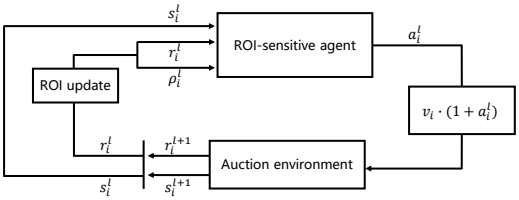

Figure 2: Illustration of bid process based on RSDRL

6 IMPLEMENTATION DETAILS

In this section, we will describe some implementation details for the range of penalty in reward (i.e., λ in Equation (6)), the range of action (i.e., a_i^l in Equation (7)) and the computation of roi_i^l (Equation (2)).

Lemma 6.1. *In auction l, if $\lambda \ge \frac{p_i^l}{p_i^l - \beta_{i,\sigma_i^l}^l \cdot v_i}$, then any violation of cost-per-conversion constraint would receive a negative reward.*

Lemma 6.1 denotes that the value of λ should be greater than some threshold . Otherwise, the reward function can not decode the cost-per-conversion constraint as follows:

$$\begin{aligned} r_i(s_i^l, a_i^l) &= p_i^l - \max\{\lambda(p_i^l - \beta_{i,\sigma_i^l}^l \cdot v_i), 0\} \\ &\ge (1-\lambda)p_i^l + \lambda\beta_{i,\sigma_i^l}^l \cdot v_i \end{aligned}$$

The inequality holds if $p_i^l \ge \beta_{i,\sigma_i^l}^l \cdot v_i$. For example, if λ is less than 1, reward would always be positive, which may lead to the violation of cost-per-conversion constraint. So with a carefully selected λ, this reward would be negative in each auction if the cost-per-conversion constraint is broken. One concern might be that the cost-per-conversion constraint is used to guarantee that the average cost-per-conversion is no greater than v_i, so we do not need to meet this constraint in each auction. A mild penalty method may be better. However, the data (Table 1 in Section 4) told us that $\beta_{i,\sigma_i^l}^l$ is always overvalued so we may not meet this cost-per-conversion constraint actually even reward is always positive in each auction. What's more, even if $\beta_{i,\sigma_i^l}^l$ is accurate, we can boost its value so this form of penalty still works.

Lemma 6.2. *In RSDRL, $a_i^l < 0$ is strictly dominated by $a_i^l = 0$.*

Lemma 6.2 provides the intuition that a_i^l is at least 0 in experiments. To meet the cost-per-conversion constraint, auctions can be divided into two categories: auctions where $p_i^l \le \beta_{i,\sigma_i^l}^l \cdot v_i$ and auctions where $p_i^l > \beta_{i,\sigma_i^l}^l \cdot v_i$. As for those auctions in the first category,

the payments of those auctions satisfy the cost-per-conversion constraint naturally; While for those auctions in the second category, it is still reasonable to win a part of them depending on roi_i^l. Based on the allocation rule and payment rule of VCG (Equation (3) and 4), the payment is always less than $\beta^l_{i,\sigma_i^l} \cdot b_i^l$. Hence $a_i^l < 0$ only decreases the number of winning auctions that belong to the first category.

As for roi_i^l, it is not hard to imagine that:

- It is not well-defined at the very beginning: Suppose that the first conversion event happens in auction l, then $\sum_{\hat{l}=1}^{l-1} \alpha_i^{\hat{l}}$ equals 0 which makes roi_i^{l-1} meaningless.
- It is not sensitive to some cost-per-conversion cases: Suppose that there is a case where roi_i^l equals 1 and $\sum_{\hat{l}=1}^{l} \alpha_i^{\hat{l}} \to \infty$, then the denominator of roi_i^l is huge. Therefore for the next few conversions which cost a little, the ROI would always equal to about 1 even the average cost of these conversions is small.

To overcome those difficulties, we use the k-ROI sensitive agent during implementation. To simplify notations, we still use roi_i^l to denote this as Equation (13) shows.

$$roi_i^l = \frac{\sum_{\hat{l}=t_k^l}^{l} p_i^{\hat{l}}}{k \cdot v_i} \tag{13}$$

Here $t_k^l = \min\{\bar{l} | \lfloor \frac{\sum_{\hat{l}=1}^{\bar{l}} \alpha_i^{\hat{l}}}{k} \rfloor + k > \lfloor \frac{\sum_{\hat{l}=1}^{l} \alpha_i^{\hat{l}}}{k} \rfloor\}$. In a word, we divide the auction stream into intervals. Each interval may have different length while contains the same number (i.e., k) of conversions.

7 EXPERIMENTS

Our methods are tested by both offline evaluation and online evaluation on a large e-commerce platform with real advertisers and auctions. The dataset will be introduced at first. Then for the offline evaluation, the reproduction of VCG will be introduced and we also quantitatively compare our methods with two state-of-the-art methods and a baseline. Finally, we will present the results of online evaluation when several OCPM advertisers uses RSDRL with a standard AA/BB test configuration.

7.1 Dataset

We randomly select 1000 OCPM advertisers in the large e-commerce platform as training set. Each advertiser participates in over 10 million auctions per day. All auction instances which these advertisers participate in are extracted from the log for seven days of late June, 2019. Each auction instance contains the following information (Table 2).

Table 2: The information contained within each instance

Number	Name	Description
1	predicted features	i.e., pctr, pcvr
2	ad features	i.e., pricing method, content

In order to evaluate the effectiveness of RSDRL, we also select another 10 OCPM advertisers and their involved auctions as test set.

7.2 The Reproduction of VCG

We obtained the simulated CTR and CVR at hour-aggregation level from the auction log, i.e., clicks within an hour is divided by views in that hour to get the simulated CTR in that hour. Then using the simulated CTR and CVR, we simulate the click and conversion events in our experiments.

We also extract the necessary ingredients of their competitors to simulate the auction environment in each auction instance, e.g., the competitors' bids. These ingredients would NOT be used as features in our training. Then based on Equation (3) and 4, we can reproduce the VCG auction for our experiments.

7.3 Compared methods

The compared methods in our experiments include:

- Truthful bidding: In each auction, OCPM advertiser i always truthfully bid (i.e., v_i). We treat this algorithms as the fundamental baseline of the experiments.
- LADDER [23]: LADDER is a deep reinforcement learning algorithm that can successfully learn bidding policies on real data. They aim at increasing the revenue of platform while also decreasing the cost-per-click of advertisers at the same time. In order to do so, they sample an auction in every 100 auctions interval as the next state. And they use net profits of every auction as rewards (i.e., $\beta^l_{i,\sigma(i)} v_i - p_i^l$ in auction l).
- SS-RTB [27]: SS-RTB is also a deep reinforcement learning algorithm for bidding optimization in sponsored search auction. They try to minimize the cost-per-conversion while guarantee a certain amount conversions. As in [27], they observe that auction sequences of two days share similar transition patterns at a proper aggregation level. Thus, they formulate the model at hour-aggregation level of the auction data. And the reward function is designed in terms of conversions.

In our experiment, there are 178 features, including advertiser-related features, auction-related features, network flow predication and so on. The continuous features are normalized and the discrete features are embedded with one-hot encoding. We use a deep model with 3 fully connected layers. The value of λ is 5.7 and the maximum value of a_i^l is 2, i.e., the generated bids are less than 3 times of their target cost-per-conversion. All algorithms share the same network architectures and the same provided features.

7.4 The offline evaluation

Figure 3 denotes the payment results of these 4 methods. The x-axis represents the training epochs and the y-axis represents the payment of OCPM advertisers in testing data. We use the payment as a critical metric because of Theorem 3.1. As Figure 3 reveals, our algorithm is better than the others in terms of payment. This experiment verifies the effectivity of our algorithm. It is worth noting that all RL methods is better than the baseline (i.e., always truthfully bid). The reason is that OCPM advertiser wins more auction if he overbids appropriately. Since the payment is always less than his

bid based on the VCG auction, overbidding is a good strategy for OCPM advertiser. Besides, our algorithm is significantly better than the other two RL based algorithms from the very beginning. The reason is that for OCPM advertiser i, the bid in our algorithm is the product of two parts (Equation (7)), i.e., the base-bid part v_i and the adjustment part $1 + a_i^l$. As a result, the bid is more than v_i. While in the other RL based algorithms, they generate bid directly which can be far less than v_i. Thus, our algorithm can get better results at the very beginning. We verify the effectivity of this bid generation method by experiments, as shown in Figure 5.

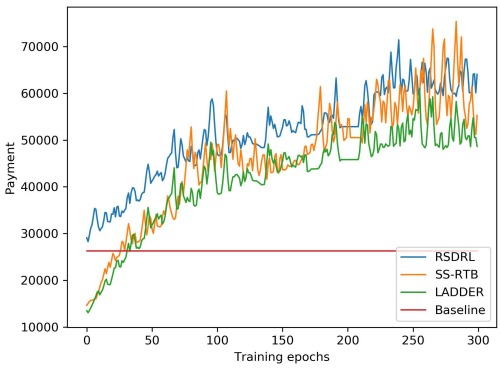

Figure 3: The payment comparisons of four algorithms

Figure 4 implies the conversion results of these four methods. The x-axis represents the training epochs and the y-axis represents the conversion number of OCPM advertisers in testing data. Based on this figure, all reinforcement learning method is better than the baseline. And these results is in accordance with those in Figure 3 which implies the effectivity of our reward function. We think this improvement is significant actually. It doubles the CONVERSIONS of OCPM advertisers on industry real testing data, in other words, it helps those OCPM advertisers double their revenue. Therefore, we conclude reinforcement learning is useful in this optimization and RSDRL has a good performance on real industry data.

In Figure 5, we compare the payment results of the following algorithms.

- Baseline:Truthful bidding.
- Base_bid: This is exactly our algorithm.
- Non-base_bid: This is also our algorithm except that the bid is generated directly, that is, directly learning bid b_i^l instead of learning a_i^l to generate bid $v_i(1 + a_i^l)$.

The x-axis represents the training epochs and the y-axis represents the payment of OCPM advertisers in testing data. Based on the experiment results, Base_bid is better than Non-base_bid, especially at the very beginning. This verifies our assumption that the value of bid is not instructive. What counts is the ratio of bid to cost-per-conversion target in the complicated dynamic environment. Therefore, decomposing bid generation into two parts like Equation (7) can have better performance than generating bid directly. What's more, this proposed bid generation methods also provides a good start point. With a carefully selected range of a_i^l (based on Lemma 6.2), it can outperform baseline from the very beginning.

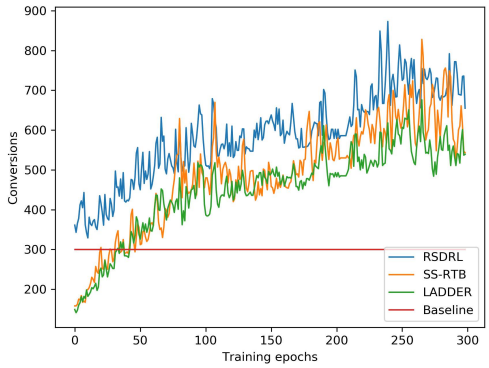

Figure 4: The conversion comparisons of four algorithms

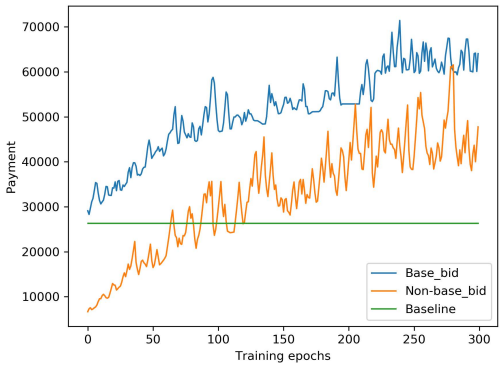

Figure 5: The payment comparisons of different action spaces

In Figure 6, the effectivity of ROI-sensitive agent is also verified. The x-axis represents the training epochs and the y-axis represents the ratio of total payment to total value, i.e., Equation (14).

$$ratio = \frac{\sum_{l=1}^{|\mathcal{I}_i|} p_i^l}{v_i \sum_{l=1}^{|\mathcal{I}_i|} \alpha_i^l} \tag{14}$$

To meet the cost-per-conversion constraint in Equation (1), the ratio should be less than 1. The orange line denotes the result for the ROI-neutral agent, i.e., $\rho : x \longrightarrow x$. While the blue line denotes the result for the ROI-sensitive agent. As Figure 6 shows, our algorithm can meet the cost-per-conversion constraint by using ROI-sensitive agent.

Figure 7 denotes the increased revenue of platform compared with baseline. The x-axis represents the training epochs and the y-axis represents the increased revenue of platform. It shows that maximizing the objective of OCPM advertiser is a win-win methods for both advertisers and platform, that is, it can increase the number of conversions for advertisers and increase the revenue of platform simultaneously. The intuition of the increased revenue comes from two aspects:

- OCPM advertisers can win more auctions, and for these auctions, they have to pay more compared to the original winner by using RSDRL.

- OCPM advertisers bids higher in those auctions where they still lose. For these auctions, the winner has to pay more based on the payment rule of VCG (Equation (4)).

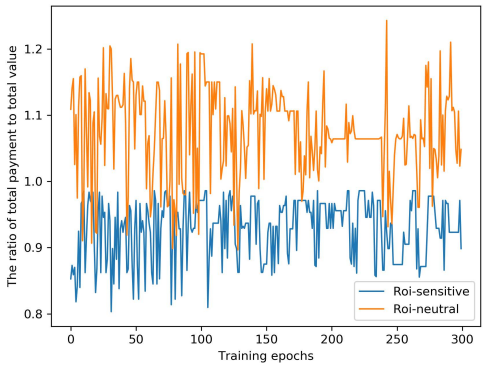

Figure 6: The ratio of total payment to total value

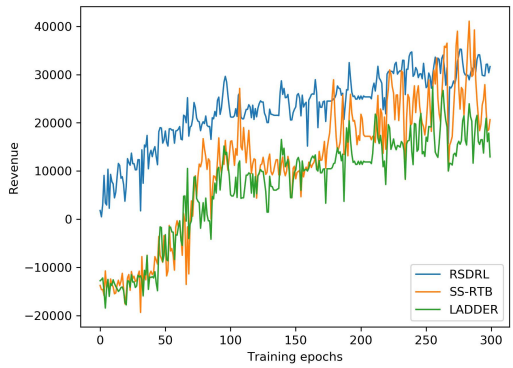

Figure 7: The increased revenue of platform compared with baseline

Figure 8 denotes the part of increased revenue of platform compared with baseline where OCPM advertisers force the winner to pay more. The x-axis represents the training epochs and the y-axis represents the corresponding increased revenue of platform. This part consumes a SMALL proportion of increased revenue when compared with Figure 7. It only takes less than 10 percent share of the total. We think this is a POSITIVE result actually since advertisers with other pricing method would not feel like they have to pay more when the platform tries to maximize the objective of OCPM advertiser.

7.5 The online evaluation

This section presents the results of online evaluation in the real-world auction environment where multiple OCPM advertisers have adopted RSDRL. We use the AA/BB-test to testify the effectiveness, as Table 3 shows. There are 4 columns denoting 4 groups. Groups A_1 and A_2 are the results of method which is adopted by the platform. Groups B_1 and B_2 are the results of RSDRL.

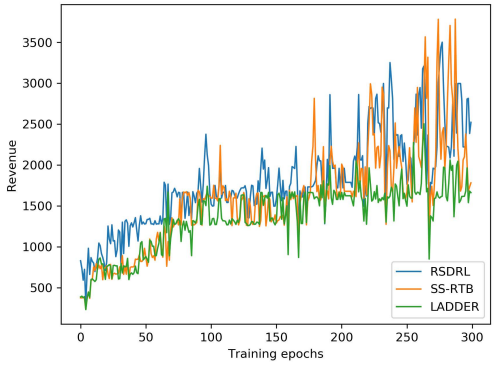

Figure 8: The part of increased revenue of platform compared with baseline caused by the highest lost bid is bidden up

By using AA/BB-test instead of A/B-test, we can reduce the risk that B would win just by chance. We sum up the results of 3 days in October, 2019, thus we can also reduce the risk of fluctuation in different days. The value of a_i^l is less than 2, i.e., we cannot bid over 3 times as many as the cost-per-conversion target. As for conversions, both B_1 and B_2 outperform A_1 and A_2, they improve the conversions for about 5%; As for payment, both B_1 and B_2 also outperform A_1 and A_2, they improve the payment more than 12%; As for ROI, all of them is less than 1, hence the cost-per-conversion constraint is satisfied.

Table 3: The results of online evaluation when multi OCPM advertisers use RSDRL

	A_1	A_2	B_1	B_2
conversions	3058	3026	3213	3189
payment	162329	160176	180875	192263
ROI	0.80	0.81	0.87	0.93

In a word, it is convincing to say that RSDRL can handle the multiple OCPM advertisers.

8 CONCLUSION

In this paper, we study the bid optimization for OCPM advertiser in feeds and provide a distributional reinforcement learning framework, named RSDRL. By making full use of the characteristics of VCG auction, we design a reward function to surrogate conversion events and a bid generation method based on theoretical results. We also provide some theoretical results to guide hyperparameter tuning. Plenty of experiments are used to evaluate the performance of our framework and RSDRL yields substantially better results than compared algorithms.

ACKNOWLEDGE

This work is supported by Science and Technology Innovation 2030 –"New Generation Artificial Intelligence" Major Project No. 2018AAA0100904, 2018AAA0101103, Turing AI Institute of Nanjing and Beijing Academy of Artificial Intelligence(BAAI) and National Natural Science Foundation of China (Grant No. 61806121).

REFERENCES

[1] Kareem Amin, Michael J. Kearns, Peter Key, and Anton Schwaighofer. 2012. Budget Optimization for Sponsored Search: Censored Learning in MDPs. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*. 54–63. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2269&proceeding_id=28

[2] Marc G. Bellemare, Will Dabney, and Rémi Munos. 2017. A Distributional Perspective on Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. 449–458. http://proceedings.mlr.press/v70/bellemare17a.html

[3] Niv Buchbinder, Kamal Jain, and Joseph Naor. 2007. Online Primal-Dual Algorithms for Maximizing Ad-Auctions Revenue. In *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*. 253–264. https://doi.org/10.1007/978-3-540-75520-3_24

[4] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R. Devanur. 2011. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*. 1307–1315. https://doi.org/10.1145/2020408.2020604

[5] Edward H. Clarke. 1971. Multipart Pricing of Public Goods. *Public Choice* 11, 1 (1971), 17–33.

[6] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018. Implicit Quantile Networks for Distributional Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. 1104–1113. http://proceedings.mlr.press/v80/dabney18a.html

[7] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. 2018. Distributional Reinforcement Learning With Quantile Regression. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 2892–2901. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17184

[8] Esther David, Rina Azoulay-Schwartz, and Sarit Kraus. 2003. Bidders' strategy for multi-attribute sequential english auction with a deadline. In *The Second International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2003, July 14-18, 2003, Melbourne, Victoria, Australia, Proceedings*. 457–464. https://doi.org/10.1145/860575.860649

[9] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* 97, 1 (2007), 242–259.

[10] Eyal Even-Dar, Vahab S. Mirrokni, S. Muthukrishnan, Yishay Mansour, and Uri Nadav. 2009. Bid optimization for broad match ad auctions. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*. 231–240. https://doi.org/10.1145/1526709.1526741

[11] Jon Feldman, S. Muthukrishnan, Martin Pál, and Clifford Stein. 2007. Budget optimization in search-based advertising auctions. In *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*. 40–49. https://doi.org/10.1145/1250910.1250917

[12] Theodore Groves. 1973. Incentives in Teams. *Econometrica* 41, 4 (1973), 617–631.

[13] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. 2018. Rainbow: Combining Improvements in Deep Reinforcement Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. 3215–3222. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17204

[14] Thomas Nedelec, Noureddine El Karoui, and Vianney Perchet. 2019. Learning to bid in revenue-maximizing auctions. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 4781–4789. http://proceedings.mlr.press/v97/nedelec19a.html

[15] Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, and Yong Yu. 2019. Deep Landscape Forecasting for Real-time Bidding Advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 363–372. https://doi.org/10.1145/3292500.3330870

[16] Weiran Shen and Pingzhong Tang. 2017. Practical versus optimal mechanisms. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 78–86.

[17] Weiran Shen, Pingzhong Tang, and Yulong Zeng. 2018. Buyer-Optimal Distribution. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1513–1521.

[18] Weiran Shen, Pingzhong Tang, and Yulong Zeng. 2018. A Closed-Form Characterization of Buyer Signaling Schemes in Monopoly Pricing. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (Eds.). International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 1531–1539. http://dl.acm.org/citation.cfm?id=3237928

[19] Weiran Shen, Pingzhong Tang, and Yulong Zeng. 2019. Buyer Signaling Games in Auctions. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, 1591–1599. http://dl.acm.org/citation.cfm?id=3331878

[20] Pingzhong Tang, Zihe Wang, and Xiaoquan (Michael) Zhang. 2016. Optimal Commitments in Asymmetric Auctions with Incomplete Information. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, Vincent Conitzer, Dirk Bergemann, and Yiling Chen (Eds.). ACM, 197–211. https://doi.org/10.1145/2940716.2940739

[21] William Vickrey. 2012. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance* 16, 1 (2012), 8–37.

[22] Jun Wang, Weinan Zhang, and Shuai Yuan. 2017. Display Advertising with Real-Time Bidding (RTB) and Behavioural Targeting. *Foundations and Trends in Information Retrieval* 11, 4-5 (2017), 297–435. https://doi.org/10.1561/1500000049

[23] Yu Wang, Jiayi Liu, Yuxiang Liu, Jun Hao, Yang He, Jinghe Hu, Weipeng P. Yan, and Mantian Li. 2017. LADDER: A Human-Level Bidding Agent for Large-Scale Real-Time Online Auctions. *CoRR* abs/1708.05565 (2017). arXiv:1708.05565 http://arxiv.org/abs/1708.05565

[24] Zihe Wang and Pingzhong Tang. 2014. Optimal mechanisms with simple menus. In *ACM Conference on Economics and Computation, EC '14, Stanford , CA, USA, June 8-12, 2014*, Moshe Babaioff, Vincent Conitzer, and David A. Easley (Eds.). ACM, 227–240. https://doi.org/10.1145/2600057.2602863

[25] 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, Sydney, NSW, Australia, August 10-13, 2015*. 1305–1314. https://doi.org/10.1145/2783258.2783276

[26] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*. 1077–1086. https://doi.org/10.1145/2623330.2623633

[27] Jun Zhao, Guang Qiu, Ziyu Guan, Wei Zhao, and Xiaofei He. 2018. Deep Reinforcement Learning for Sponsored Search Real-time Bidding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 1021–1030. https://doi.org/10.1145/3219819.3219918

[28] Han Zhu, Junqi Jin, Chang Tan, Fei Pan, Yifan Zeng, Han Li, and Kun Gai. 2017. Optimized Cost per Click in Taobao Display Advertising. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. 2191–2200. https://doi.org/10.1145/3097983.3098134