# Reinforcement Mechanism Design: With Applications to Dynamic Pricing in Sponsored Search Auctions*

**Weiran Shen[1], Binghui Peng[1], Hanpeng Liu[1], Michael Zhang[2], Ruohan Qian[3], Yan Hong[3], Zhi Guo[3], Zongyao Ding[3], Pengjun Lu[3], Pingzhong Tang[1]†**

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University

[2]Chinese University of Hong Kong

[3]Baidu Inc.

## Abstract

In many social systems in which individuals and organizations interact with each other, there can be no easy laws to govern the rules of the environment, and agents' payoffs are often influenced by other agents' actions. We examine such a social system in the setting of sponsored search auctions and tackle the search engine's dynamic pricing problem by combining the tools from both mechanism design and the AI domain. In this setting, the environment not only changes over time, but also behaves strategically. Over repeated interactions with bidders, the search engine can dynamically change the reserve prices and determine the optimal strategy that maximizes the profit. We first train a buyer behavior model, with a real bidding data set from a major search engine, that predicts bids given information disclosed by the search engine and the bidders' performance data from previous rounds. We then formulate the dynamic pricing problem as an MDP and apply a reinforcement-based algorithm that optimizes reserve prices over time. Experiments demonstrate that our model outperforms static optimization strategies including the ones that are currently in use as well as several other dynamic ones.

## Introduction

Selling advertisements online through sponsored search auctions is a proven profit model for Internet search engine companies such as Google and Baidu. When a user submits a query in such a search engine, it displays, in the result page, a few advertisements alongside the organic results, both related to the query. In the backend, the keyword search triggers an auction mechanism among all advertisers who are interested in the keyword. The advertisers submit bids to compete for advertising positions on the result page. The search engine then ranks the advertisements on the result page according to the advertisers' bids and charges them only when someone clicks on the advertisement.

*This work is supported by Science and Technology Innovation 2030 - "New Generation Artificial Intelligence" Major Project No.(2018AAA0100904) and Beijing Academy of Artificial Intelligence (BAAI). An early version of this paper was posted on arxiv.org (Shen et al. 2017).

†kenshinping@gmail.com. To whom correspondence should be addressed.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

The gold-standard mechanism in sponsored search is the well-known *generalized second price (GSP) auction* (Edelman, Ostrovsky, and Schwarz 2007; Varian 2007). The auctions allocate the best slots to the advertisers who submit the highest bids, second best slots to the ones with the second highest bids, and so on; and charge them based on the bids one slot below them (or the lowest price for them to maintain the current slot). Major search engines all adopt some variants of the GSP auction.

A problem with the vanilla GSP auction is that it is not revenue optimal, according the seminal theory attributed to Myerson (1981). It is known that, under standard game theory assumptions, a revenue-optimal auction does not necessarily allocate the slots by the rank of their bids. It is also known that in an optimal auction, there exists a vector of advertiser-specific reserve prices that filter low bids. Over the years, a large body of literature at the interface of the economics and computer science has focused on revenue optimization of GSP auctions by incorporating insights (ranking and reserve price) from Myerson's theory (Lahaie and Pennock 2007; Roberts et al. 2013; Ostrovsky and Schwarz 2011; Thompson and Leyton-Brown 2013).

However, most revenue optimization theories depend crucially on the assumptions about the bidders' rational behaviors. Recently, it has been shown that such assumptions may not hold in reality. Therefore, an emerging line of works has started to focus on settings where the bidders use a certain learning algorithm (Nekipelov, Syrgkanis, and Tardos 2015; Balseiro and Gur 2017; Nazerzadeh et al. 2016). However, most of these models do not give a specific and detailed description of the bidders' actual behaviors. Also, the different rationality levels of the heterogeneous bidder population cannot be easily captured.

### Our approach: reinforcement mechanism design

Instead of applying the mechanism design theory or machine learning techniques in isolation, we propose a hybrid approach that combines insights from both domains to examine how a social system with human players can be better designed. First, we follow the mechanism design theory and describe our mechanism with a set of parameters. Then we apply optimization algorithms to search for the optimal pa-

rameters. Meanwhile, in order to tackle the so-called "second-order" effect (bidders have different behaviors under different mechanisms), we use machine learning algorithms to build a precise bidder behavior model that explicitly takes mechanism parameters as inputs. Our framework takes into account both machine learning and strategic bidder behaviors.

The first part of this paper tries to solve the problems mentioned above by building an end-to-end neural network based bidder behavior model. Our model consists of both a public feature set and a private feature set, and directly predicts a bidder's bidding behaviors using the features that are observable by the bidder. Our model scales well with extremely large dataset and can also handle heterogeneous bidders due to the great flexibility of neural networks. We formulate our bidder behavior model mathematically as a Markov model.

In the second part of the paper, enabled by the Markov model formulation, we can view the dynamic mechanism design problem as a Markov decision process (MDP). We then solve the MDP with reinforcement learning techniques. Specifically, our objective is to design an optimal mechanism so the performance of the system can be improved through the change of some policy parameters even in the presence of strategic players in the system. This modification of the setting creates several significant challenges. First, the system is not static any more. In each round, the algorithm will attempt to change some policy parameter for optimal mechanism design. This changes the environment for the players. Second, at the same time when the rule is changing, the players' strategies change too. They can adopt complex strategies to react to changing rules. This moving-target nature of the setting makes it difficult to implement the optimization. Third, as in the traditional setting, reinforcement learning requires large amount of feedbacks for the training, but tweaking the environment is usually very costly.

Our approach is also clearly different from classic mechanism design theory. We relax some unrealistic assumptions such as quasi-linear utility and bidders' rationality. Instead, we utilize machine learning to learn bidder behaviors.

We solve these problems in the setting of sponsored search auctions and explore the use of an AI-driven mechanism so that search engines can dynamically set minimum bid prices and use the data generated in the process to maximize the profit. The proposed algorithm can be generalized and applied to other settings to improve system design.

## Our contributions

We make two major contributions in this paper:

- We propose a neural network based bidder behavior model. Our model follows the rich research literature of behavioral economics (Wright and Leyton-Brown 2017; 2014). Our choice of RNN is commonly used to deal with time series data. Based on this, we also provide a Markov behavior interpretation, which enables us to use tools from other domains to search for a dynamic mechanism with good enough performance.
- Based on the above bidder behavior model, we model the dynamic mechanism design problem as a Markov decision process, and use the Monte-Carlo Tree Search algorithm to find a dynamic mechanism that has much better performance than the current one online.

A version of our framework has already been implemented in the online ad auction system in Baidu, and has been proven to be able to significantly increase the revenue (see Baidu's Financial Report of Q1 2018 (2018)).

## Related works

In the AI community, a recent, interesting line of works aims to tackle the revenue optimization problem from a learning perspective. For example, Duetting et al. (2019) and Shen, Tang, and Zuo (2019) aim to learn optimal mechanisms via neural networks. In the dynamic auction literature, Mohri and Medina (2015) and Mohri and Medina (2016) apply learning algorithms to exploit past auctions and user features. Their algorithms rely on the implicit assumption that buyers do not change their behaviors over time. Mohri and Munoz (2015) and Mohri and Munoz (2014) aim to maximize revenue with strategic buyers and give desirable regret bounds on online pricing algorithms. These works assume that each buyer has an underlying bid or value distribution and it does not change over time. There are also a series of works that optimize the revenue in dynamic auctions with the so-called "bank account" mechanisms (Mirrokni et al. 2016a; 2016b; Shen, Wang, and Zuo 2018).

Battaglini (2005) study the Markovian consumer model in a long-term contract setting. Their results show that even when the types at different times are highly persistent, the optimal contract is far from a static one. He et al. (2013) and Tian et al. (2014) also assume that buyers have the Markov property and the goal of Tian et al. (2014) is to find the best static mechanism.

One objective of works in the literature of sponsored search auctions is to improve the revenue of GSP auctions (Lahaie and Pennock 2007; Thompson and Leyton-Brown 2013; Shen and Tang 2017). When designing and analyzing these auctions, most of these works make the standard game-theoretical assumption that advertisers have a single parameter called *type* that indicates their maximum willingness-to-pay for a single click-through. During evaluation, these works also assume that advertisers are rational and will play according to some equilibrium.

While these works shed lights on how to design sponsored search auctions in theory, the assumptions they make do not generally hold in practice. Most advertisers have complex private information, such as budget constraints (Xu et al. 2013; Balseiro and Gur 2017), multi-dimensional valuation, and negative externalities (Jehiel, Moldovanu, and Stacchetti 1996; Deng and Pekec 2011). Furthermore, private information such as budget may change dynamically over time and advertisers may not be able to observe all configuration parameters of the auction.

There are a few exceptions in the literature that take the initiative to design and evaluate sponsored auctions by getting rid of these assumptions. Ostrovsky and Schwarz (2011) conduct large field experiments by manually setting different levels of reserve prices. They show that by incorporating

discounted Myerson's reserve prices, the search engine can improve its revenue. However, it remains unclear about the long-term performance of these auctions since all these auctions are assumed to be static. It is also unclear how the *ad hoc* selection of the reserve prices can be improved. Nekipelov, Syrgkanis, and Tardos (2015) investigate the problem of estimating the valuations of the bidders from their bids. They get rid of the assumption that the bidders bid according to some equilibrium and make a milder assumption that bidders play according to some no-regret learning strategy.

Deep reinforcement learning methods have successfully produced AI agents that can beat human players (Mnih et al. 2015; Silver et al. 2016). A recent paper by Racanière et al. (2017) proposes "imagination-augmented agents" and applies the method to a Sokoban game where the player needs to move boxes to given target locations. With a pre-trained model based on simple levels, the AI agent can solve more difficult levels, demonstrating interesting learning capabilities. Deep reinforcement learning also showed powerful potential of developing control policies in physical systems. For example, Tai, Paolo, and Liu (2017) report that models trained in a simulator can be adopted by real robots. In all these settings, the environment is given and the agents' payoffs are easily determined based on the rules of the environment. Beyond these applications, deep reinforcement learning has also been applied to other economic settings such as e-commerce platforms (Cai et al. 2018).

# Preliminaries

## Sponsored search auction and Baidu's design

We consider an auction design problem in the sponsored search setting. When a user types a keyword query in a search engine, the search engine (called the seller hereafter) displays, in the result page, a few advertisements related to the keyword. We consider auctions of a single keyword, with $N$ bidders competing for $K$ slots. Each bidder $i$ reports a bid $b_i$ to the seller. A bid profile is denoted by $b = (b_1, b_2, \ldots, b_N)$. We slightly abuse notations and use $b_i$ to refer to both bidder $i$ and his bid.

In a standard game-theoretical model, there is a single-dimensional type for each bidder that denotes the maximum amount of money that the bidder is willing to pay. However, we do not explicitly emphasize such a value in our model. The reason is two-fold: first, our model does not assume that the bidders are fully rational or rational according to some metric. Second, there are many factors that may affect bidders' bidding behavior, so explicitly define one such parameter that we cannot observe does not help much in end-to-end training. These are also the reasons why our bidder behavior model is defined over, instead of their private information, the bidders' observations and past bidding data. In fact, this kind of data-driven model is not uncommon in the literature (He et al. 2013; Xu et al. 2013; Pin and Key 2011).

In this paper, we attempt to relax the unrealistic assumptions and consider an environment in which bidders can have arbitrarily complex private information and arbitrary rationality levels that can change dynamically over time. Our goal is to design dynamic mechanisms that yield competitive revenue in practice in the long run. While the framework and algorithms proposed in this paper are applicable to search engines in general, we focus on the sponsored search auction design of Baidu, the largest search engine in China. We use Baidu as a running example throughout the paper, calibrating our model with its data.

Baidu sells 3 ad slots for most keywords and like other major search engines, Baidu runs a type of randomized, GSP-like auction mechanism to sell the slots. The bidding data yielded by the randomness of the mechanism provides a perfect setting for us to learn how bidders react to different choices of reserve prices and the number of impressions, and the induced click-through-rates (CTRs).

## The GSP mechanism

Upon receiving a search query, the seller needs to determine a slot allocation and payment vector. Formally, a *mechanism* consists of two functions $\mathcal{M} = (x, p)$, where the *allocation rule* $x$ is a function $x : \mathbb{R}^N \to [0,1]^N$, which takes as input the bid profile and outputs an $N$-dimensional vector indicating the quantity of items allocated to each bidder; and the payment rule $p$ is a function $p : \mathbb{R}^N \to \mathbb{R}^N$ that maps the bid profile to an $N$-dimensional non-negative vector specifying the payment of each bidder.

We consider the GSP (generalized second price) auction that are widely adopted by major search engines. Suppose there are $N$ bidders competing for $K$ advertising slots. The $K$ slots have different effects of attracting user clicks (described by their CTRs). Denote by $q_k$ the CTR of the $k$-th slot and assume that $q_k$ is non-increasing with respect to the position of the slot, i.e. $q_1 \geq q_2 \geq \cdots \geq q_K \geq 0$. Upon receiving a keyword query, the seller first collects the bid profile $b$ from the bidders. Usually, each bidder is associated with a reserve price $r_i$, which is the minimum quantity that bidder $i$ needs to bid in order to enter the auction. Denote by $b_{(i)}$ the $i$-th highest bid among those above the reserve prices. The seller then sequentially allocates the $i$-th slot to bidder $b_{(i)}$, until either the slots or the bidders run out. When bidder $b_{(i)}$'s advertisement is clicked by a user, the seller charges the bidder according to the following rule:

$$p_{(i)} = \begin{cases} \max\left\{\frac{q_{i+1} b_{(i+1)}}{q_i}, r_{(i)}\right\} & \text{if } b_{(i+1)} \text{ exists;} \\ r_{(i)} & \text{otherwise.} \end{cases}$$

The reserve price profile $r$ can significantly affect the revenue of the advertising platform. In this paper, we view the reserve price profile $r$ as the main parameters of the mechanism. The seller's goal is to set reserve price profiles dynamically to maximize its revenue.

# Bidder behavior model

The mechanism design theory relies crucially on how the bidders behave. Classical game theoretical analysis depends on the following assumptions:

- the bidders have quasi-linear utility;
- the bidders have unlimited information access and computational power to compute a Nash equilibrium.

However, these assumptions become problematic in the real world. First, different bidders may have advertising campaigns with different objectives. For example, a bidder who wants to increase the awareness of his brand may only care about the number of impressions, while a budget-constrained bidder who aims to increase the sales volume may focus on the number of clicks of his advertisement in a specific slot. Second, in real advertising platforms, the bidders can only access information about their own advertisements. Empirical evidence has also shown that the above assumptions may not hold in sponsored search auctions (Edelman and Ostrovsky 2007; Pin and Key 2011).

### RNN-based bidder model

In our model, each bidder's action is his bid distribution. The reason why the bids forms a distribution is that a bidder may place different bids for different user characteristics. Each bidder $i$ is a function $g_i$ that takes as input the history bid distributions and his KPIs (key performance indicators), and outputs the bidder's bid distribution of the next time step. To fit these time series data, we use a standard Long Short-Term Memory (LSTM) recurrent neural network. The output of the RNN is further transformed through a common fully connected with a softmax activation function to ensure that the final output of the network is a valid probability distribution. The inputs of the network include KPIs of $m$ consecutive days, the bid distributions for the bidder and also some date related features (summarized in Table 1).

Table 1: List of features

| Feature | Representation |
|---|---|
| bid distribution | 100-dimensional vector |
| #impressions from different slots | tile-coding of logarithm value |
| #clicks | |
| total payment | |
| month of campaign season | one-hot encoding |
| day of month | |
| day of week | |

To simplify the representation of the bid distribution, we discretize the with 100 non-overlapping intervals and use a 100-dimensional vector $b$ to describe a bid distribution. These 100 intervals are computed according to history data so that each interval contains roughly the same number of bids placed by all bidders.

Our choice of KPI statistics for each bidder includes the number of impressions the bidder obtains from each slot, the total number of obtained clicks and the total amount of payment. Our observation in Baidu shows that the bidders care more about relative changes of their KPIs rather than absolute changes. For example, an increase of 100 clicks makes no difference at all for a bidder obtaining 2 million clicks every day, but can be quite significant for a small bidder obtaining 200 daily clicks. Therefore, to capture such relative changes, we use the logarithm value of these KPI statistics as the input feature in our RNN and encode them with tile-coding.

Besides the above private features, we also include a public feature set, including date related features such as the month and the day of the week. All of these features are encoded with one-hot encoding. The reason for including these features is that most advertisers have seasonal advertising campaigns and may adjust their bidding strategies according to the current date.

### Mathematical formulation: Markov bidder model

Similar to (He et al. 2013; Xu et al. 2013), we adopt the time-homogeneous Markov model to interpret the RNN-based bidders' behavior model. Denote by $s_i^{(t)}$ and $h_i^{(t)}$ the bid distribution of bidder $i$ and the KPIs received by bidder $i$ at time step $t$. The bidders may adjust their bids dynamically according to their KPIs. Thus the bid distribution of bidder $i$ at the next time step is a function of previous $s_i$'s and $h_i$'s:

$$s_i^{(t+1)} = g_i\left(s_i^{(t-m+1:t)}, h_i^{(t-m+1:t)}\right) \quad (1)$$

where $s_i^{(t-m+1:t)}$ and $h_i^{(t-m+1:t)}$ are bidder $i$'s bid distributions and KPIs of $m$ consecutive time steps, respectively. Such a Markov model is not uncommon in the literature, see (He et al. 2013; Battaglini 2005). Our experiences with Baidu also indicate that the Markov model aligns with the bidders' behaviors.

The prediction of our network is quite accurate according to Figure 2. One might argue that the bidders may be very "lazy" and do not often change their bids, and in this case, obtaining an accuracy as shown in Figure 2 is not significant at all. However, our previous online experiment shows that the bidders actually change their bids quite frequently. We also simulated this experiment offline using only our bidder model and get very similar results as the online experiment (see the next section for details).

## The reinforcement mechanism design framework

In this section, we describe how we formulate the dynamic mechanism design problem as a Markov decision process and describe ways that we solve it.

The bids of the $N$ bidders are drawn from their bid distributions. We make the assumption that the individual bids are independent of each other. While such an assumption loses generality, it is in fact quite widely used in the literature (Mohri and Medina 2015; He et al. 2013). The joint bid distribution is

$$\begin{aligned} s^{(t+1)} &= \prod_{i=1}^{N} s_i^{(t+1)} = \prod_{i=1}^{N} g_i\left(s_i^{(t-m+1:t)}, h_i^{(t-m+1:t)}\right) \\ &= g\left(s^{(t-m+1:t)}, h^{(t-m+1:t)}\right) \end{aligned}$$

For simplicity, we assume that the number of daily queries of each keyword is a constant. Thus, the KPI $h^{(t)}$ is completely determined by both the bid distribution $s^{(t)}$ and the reserve price profile $r^{(t)}$.

Thus we can formulate the dynamic mechanism design problem as a Markov decision process, where we view $s^{(t)}$ as the state of the seller and $r^{(t)}$ as its action.

**Definition 1.** *The long-term revenue maximization problem is a Markov decision process* $(\mathcal{N}, S, R, G, REV(s, r), \gamma)$*, where*

- $\mathcal{N}$ *is the set of bidders with* $|\mathcal{N}| = N$.
- $S = S_1 \times \cdots \times S_N$ *is the state space, where* $S_i$ *is the set of all possible bid distributions of bidder* $i$*;*
- $R = R_1 \times \cdots \times R_N$ *is the action space, where* $R_i$ *is the set of all possible reserve prices that the mechanism designer can set for bidder* $i$*;*
- $G = (g_1, g_2, \ldots, g_N)$ *is the state transition functions;*
- $REV(s, r)$ *is the immediate reward function that gives the expected revenue for setting reserve price profile* $r$ *when the state is* $s$*;*
- $\gamma$ *is the discount factor with* $0 < \gamma < 1$.

**Remark 1.** *Note that although we use the revenue as the immediate reward in this particular task, we can change it to any other function without changing the framework.*

The objective is to select a sequence of reserve price profiles $\{r^{(t)}\}$ that maximizes the sum of discounted revenues:

$$OBJ = \sum_{t=1}^{\infty} \gamma^t REV\left(s^{(t)}, r^{(t)}\right).$$

Figure 1 shows the main framework of the dynamic mechanism design problem. The framework contains two parts:

1. Markov bidder model (the RNNs in our case, as described in the formal section), which determines how bidders adjust their bids according to the KPI feedbacks;
2. Mechanism, where the bidders interact with the seller's action (reserve prices) and get KPIs as feedbacks.

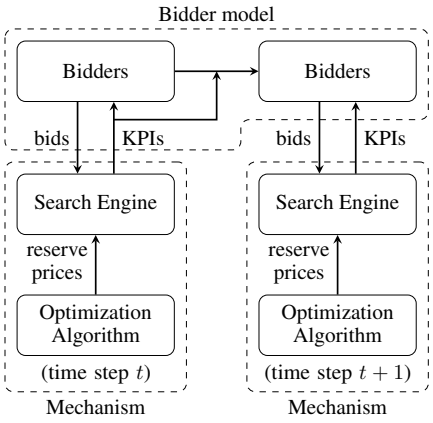


Figure 1: Model framework

## Optimization algorithm: Monte-Carlo Tree Search

Although an optimal reserve pricing scheme exists according to the MDP theory, its exact computation is formidably costly due to the following reasons:

- The possible reserve profiles grows exponentially with respect to the number of the bidders;
- The number of future states to explore is exponential with respect to the searching depth.

We circumvent the first difficulty by restricting attentions to keywords that contain only a few major bidders. We focus on the keywords with thin markets (few major bidders) mainly because the effect of reserve prices diminishes in thick markets anyway. To tackle the second one, we only explore possible actions for a bidder that are in a small neighborhood of the current reserve price. This restriction is also necessary for practical stability concerns, since sudden changes in reserve prices would result in sudden changes in bidders' KPIs, which would hurt the stability of the platform. With these restrictions, the size of the action space is greatly reduced to a small subset. To further speed up the search, we implement the *Monte-Carlo Tree Search (MCTS) algorithm* (Khandelwal et al. 2016), since the computational complexity of the MCTS algorithm can be effectively bounded by restricting the search depth and the number of search trajectories.

# Experiments

We selected 400 keywords[1] with the following properties:

- The number of daily queries for the keyword is large and stable (with small variance).
- The most part (at least $80\%$) of the revenue of the keyword is contributed by at most 3 bidders.

We extracted 8 months' bidding data related to these keywords from Baidu. The total data size of the data set is over 70TB. As mentioned before, the reason for the second condition is that the effect of reserve prices diminishes in thick markets. These 400 keywords in the data set contributes about $10\%$ of Baidu's total revenue. In our data set, each data record corresponds to an impression of an ad and contains over 300 data fields.

## Bidder behavior model

For each keyword, we only focus on the 3 major bidders and ignore others. For each bidder, we built an LSTM recurrent neural network with 128 hidden units using TensorFlow. We set the time step to be 1 day and trained it using the 8 months' data. We use the average cross entropy as the performance indicator and optimize our RNN using Tensorflow's built-in ADAM optimizer. The total data set is divided into a $90\%$ training set and a $10\%$ test set.

Recall that the input of our RNN is the bid distributions and KPIs of $m$ consecutive days. We set $m = 4$ in our experiment which has an average cross entropy of about $1.67$ among all bidders and all test instances in the test set. Some selected test instances are listed in Figure 2.

## The Monte-Carlo Tree Search algorithm

The possible reserve prices we explore for the bidder are 0.95, 1.0 and 1.05 times the current reserve price for the

[1]Our dataset is considerably larger than in most papers in the literature. For example, (Nekipelov, Syrgkanis, and Tardos 2015) conduct experiments based on 1 week's data from 9 bidders and the dataset for simulations in (Lahaie and Pennock 2007) contains only 1 keyword.

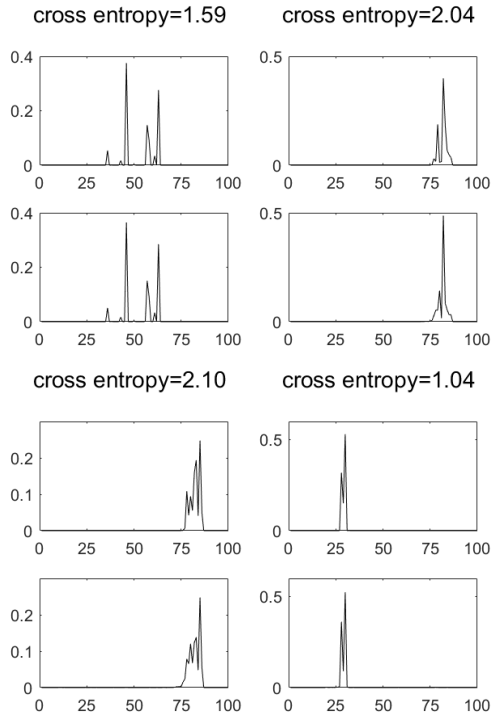


Figure 2: Prediction results for 4 selected bidders. Each subfigure contains two distributions, with the upper one being the actual distribution and the lower one being the prediction. The cross entropy of each instance is shown on top.

bidder. We set $\lambda = 0.8$ and the search depth to be 5 in our optimization algorithm. In the selection step, we restrict the number of explorations to be 5000. In the expansion step, to estimate the revenue at the selected node, we simulate the auction 5 million times and compute the average revenue as the per-impression revenue of each keyword.

We set the initial reserve price to be $p = \arg\max_b b(1 - F(b))$ where $F(b)$ is the current bid distribution. We call this reserve price *static optimal*, since this price maximizes the revenue if the bidders do not change their bids. Several algorithms are compared:

- STATIC_OPT: Always use the initial reserve.
- GREEDY: Let the revenue of the current period be $REV^t$. In each round, we randomly choose a bidder $i$ and change only his reserve price by $-5\%$ and simulate auctions for the next period. The revenue is then

$$REV_i^{t+1} = REV\left(s^{(t+1)}, \left(0.95r_i^{(t)}, r_{-i}^{(t)}\right)\right).$$

And we set

$$r^{(t+1)} = \begin{cases} \left(0.95r_i^{(t)}, r_{-i}^{(t)}\right) & \text{if } REV_i^{t+1} > REV_i^t \\ \left(1.05r_i^{(t)}, r_{-i}^{(t)}\right) & \text{otherwise} \end{cases}.$$

Notice this method can been seen as a simplified version of coordinate gradient descend (ascend) method.

- POLICY_GRAD: This algorithm is similar to apply the GREEDY algorithm to each bidder simultaneously. In this algorithm, we compute the revenue change for each bidder $i$ and change the reserve price accordingly:

$$r_i^{(t+1)} = \begin{cases} 0.95r_i^{(t)} & \text{if } REV_i^{t+1} > REV_i^t \\ 1.05r_i^{(t)} & \text{otherwise} \end{cases}.$$

- BAIDU: Current reserve prices used by Baidu.
- STATIC_50: 50 cents as the reserve prices for all bidders, regardless of bid distribution.

Note that Baidu uses randomized reserve prices in its system, while in the above algorithms, all reserve prices are deterministic. The reason of doing so is due to the company's disclosure policy.

We also compare the effect of different frequencies of changing reserve prices by setting the time step $\Delta t$ in the expansion step of the optimization algorithm[2]. Clearly, changing the reserve prices too frequently can affect the stability of the platform and thus is not desirable. In this experiment, we only compare the performance of our framework.

## Results and analysis

In the first experiment, we set $\Delta t = 1$ in our MCTS algorithm, and compare it with other strategies mentioned above. We simulate 120 days for each strategies. The results of the experiments are shown in Figure 3. Revenue is normalized with the converged value of BAIDU.

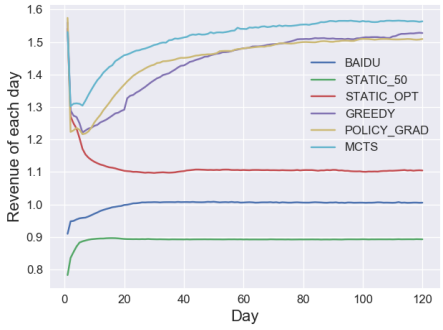


Figure 3: Performance of different strategies.

The figure shows that

- Our dynamic strategy outperforms all other static strategies including STATIC_OPT, BAIDU, STATIC_50, as well as the dynamic strategy GREEDY and POLICY_GRAD;
- The BAIDU curve converges rapidly within just few days. It goes up slightly at the beginning, mainly because our simulation uses deterministic reserve prices instead of the actual randomized ones.
- The STATIC_OPT curve undergoes a rapid rise on the first day and then followed by a steep fall, also converges after two weeks.

[2]This time step is not necessarily equal to the time step for training the Markov bidder model. We can always simulate bidder behaviors day by day but change the reserve every several days

In fact, in a previous Baidu online experiment, we set STATIC_OPT reserve prices for each (keyword, bidder) pair to test the response from the bidders. The experiment shows that setting a reserve price according to history bids that maximizes immediate revenue could result in high revenue in short term, but drops back after around 10 to 20 days. Furthermore, such a strategy increases the revenue about 10% after convergence.

In our simulations, the STATIC_OPT curve perfectly aligns with observations from the previous online experiment, which can serve as a further proof of the accuracy of our bidder behavior model.

Besides, the simulation also reveals some interesting facts about bidder behaviors:

- All aggressive pricing schemes gain high revenue immediately and drop significantly later. This phenomenon is intrinsic for our dataset, since all the bidders undergo mild pricing mechanism previously due to relatively low reserve prices and the random discounts. The sudden change in reserve price (from both adopting the static optimal reserve and discarding randomization) could make huge immediate reward, but once bidders are aware of the change and respond accordingly, less revenue can be extracted.
- Although STATIC_OPT could beat mild mechanisms like BAIDU and STATIC_50, its long term revenue is not as promising as the short term.
- The experiment shows that with more involved optimization algorithm (such as MCTS) and accurate bidder model, we could achieve the best performance and gain higher revenue in the long run.
- Surprisingly, algorithms GREEDY and POLICY_GRAD perform very well, only slightly worse than the MCTS algorithm. However, these two algorithms are much simpler and computationally cheaper. Such a result may, to some extent, suggest that the bidders are not very strategic, since simple algorithms like GREEDY can also capture their behaviors well.
- The GREEDY algorithm and the POLICY_GRAD algorithm are similar to each other, and also have similar performances. The POLICY_GRAD algorithm gives a smoother curve and converges more quickly, but the GREEDY algorithm has a slightly higher revenue when converged.

In the second experiment, we compare the effect of the frequency of changing reserve prices. The results are shown in Figure 4. We use the MCTS algorithm and also simulated 120 days for each $\Delta t$. the figure indicates that the larger $\Delta t$ is, the more revenue it can extract, and the more quickly it converges. The revenue of $\Delta t = 7$ is about several percent small than that of $\Delta t = 1$, Comparing Figure 3 and 4, we can see that the performance GREEDY algorithm is almost the same as the MCTS algorithm with $\Delta t = 3$.

### Practical implementation

One may argue that the policies explored in our experiments are too aggressive, and that using personalized reserve prices can cause fairness issues. In fact, the flexibility of our framework allows us to implement other non-aggressive policies, for example, using an anonymous reserve price, or consider other objective functions such as welfare, click yield.

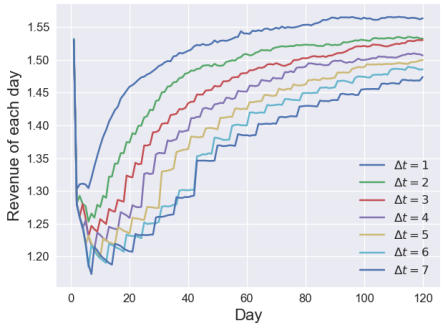


Figure 4: Effect of the frequency of changing reserve prices.

A mild version of our framework has already been implemented in the online advertising system in Baidu, and was highlighted in Baidu's Financial Report of Q1 2018 (2018).

## Conclusion

In this paper, we propose a dynamic pricing framework, that combines mechanism design with AI techniques. Our framework does not depend on unrealistic assumptions adopted by most theoretical analyses. We use a data-centric approach to solve a theoretical market-design problem.

Our framework contains two main parts: the bidder-behavior model and the optimization algorithm. The optimization algorithm finds the optimal mechanism parameters for each step iteratively. In each round, the algorithm estimates the future objectives by simulating the auctions with the bidder-behavior model.

We apply our framework to the sponsored search setting and assume Markov bidder behavior. The model uses an RNN for the bidder model and an MCTS algorithm to solve for the optimal reserve prices. Our experiments with real bidding data from Baidu, a major search engine in China, show that our framework can dramatically improve the revenue compared to other static and dynamic strategies.

Our framework has already been adopted by Baidu and is proven to be able to significantly increase revenue.

## References


Baidu Inc. 2018. First quarter 2018 financial reports. http://media.corporate-ir.net/media_files/IROL/18/188488/2018/BIDU%20--%20Q1%202018%20Earnings%20Release-d.pdf.

Balseiro, S. R., and Gur, Y. 2017. Learning in repeated auctions with budgets: Regret minimization and equilibrium. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, 609–609. ACM.

Battaglini, M. 2005. Long-term contracting with markovian consumers. *The American economic review* 95(3):637–658.

Cai, Q.; Filos-Ratsikas, A.; Tang, P.; and Zhang, Y. 2018. Reinforcement mechanism design for e-commerce. In *Proceedings of the 2018 World Wide Web Conference*, 1339–1348. International World Wide Web Conferences Steering Committee.

Deng, C., and Pekec, S. 2011. Money for nothing: exploiting negative externalities. In *Proceedings of the 12th ACM conference on Electronic commerce*, 361–370. ACM.

Duetting, P.; Feng, Z.; Narasimhan, H.; Parkes, D.; and Ravindranath, S. S. 2019. Optimal auctions through deep learning. In *International Conference on Machine Learning*, 1706–1715.

Edelman, B., and Ostrovsky, M. 2007. Strategic bidder behavior in sponsored search auctions. *Decision Support Systems* 43(1):192–198.

Edelman, B.; Ostrovsky, M.; and Schwarz, M. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American Economic Review* 97(1):242–259.

He, D.; Chen, W.; Wang, L.; and Liu, T.-Y. 2013. A game-theoretic machine learning approach for revenue maximization in sponsored search. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Jehiel, P.; Moldovanu, B.; and Stacchetti, E. 1996. How (not) to sell nuclear weapons. *The American Economic Review* 814–829.

Khandelwal, P.; Liebman, E.; Niekum, S.; and Stone, P. 2016. On the analysis of complex backup strategies in monte carlo tree search. In *Proceedings of The 33rd International Conference on Machine Learning*, 1319–1328.

Lahaie, S., and Pennock, D. M. 2007. Revenue analysis of a family of ranking rules for keyword auctions. In *Proceedings of the 8th ACM conference on EC*, 50–56. ACM.

Mirrokni, V.; Leme, R. P.; Tang, P.; and Zuo, S. 2016a. Dynamic auctions with bank accounts. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 387–393. AAAI Press.

Mirrokni, V.; Leme, R. P.; Tang, P.; and Zuo, S. 2016b. Optimal dynamic mechanisms with ex-post ir via bank accounts. *arXiv preprint arXiv:1605.08840*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Mohri, M., and Medina, A. M. 2015. Non-parametric revenue optimization for generalized second price auctions. In *Proceedings of the 31st Conference on UAI, July 12-16, 2015, Amsterdam, The Netherlands*, 612–621.

Mohri, M., and Medina, A. M. 2016. Learning algorithms for second-price auctions with reserve. *Journal of Machine Learning Research* 17(74):1–25.

Mohri, M., and Munoz, A. 2014. Optimal regret minimization in posted-price auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, 1871–1879.

Mohri, M., and Munoz, A. 2015. Revenue optimization against strategic buyers. In *Advances in Neural Information Processing Systems*, 2530–2538.

Myerson, R. B. 1981. Optimal auction design. *Mathematics of Operations Research* 6(1):58–73.

Nazerzadeh, H.; Paes Leme, R.; Rostamizadeh, A.; and Syed, U. 2016. Where to sell: Simulating auctions from learning algorithms. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 597–598. ACM.

Nekipelov, D.; Syrgkanis, V.; and Tardos, E. 2015. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 1–18. ACM.

Ostrovsky, M., and Schwarz, M. 2011. Reserve prices in internet advertising auctions: a field experiment. In *Proceedings of the 12th ACM conference on EC*, 59–60. ACM.

Pin, F., and Key, P. 2011. Stochastic variability in sponsored search auctions: observations and models. In *Proceedings of the 12th ACM conference on EC*, 61–70. ACM.

Racanière, S.; Weber, T.; Reichert, D.; Buesing, L.; Guez, A.; Rezende, D. J.; Badia, A. P.; Vinyals, O.; Heess, N.; Li, Y.; et al. 2017. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 5694–5705.

Roberts, B.; Gunawardena, D.; Kash, I. A.; and Key, P. 2013. Ranking and tradeoffs in sponsored search auctions. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, 751–766. ACM.

Shen, W., and Tang, P. 2017. Practical versus optimal mechanisms. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 78–86. International Foundation for Autonomous Agents and Multiagent Systems.

Shen, W.; Peng, B.; Liu, H.; Zhang, M.; Qian, R.; Hong, Y.; Guo, Z.; Ding, Z.; Lu, P.; and Tang, P. 2017. Reinforcement mechanism design, with applications to dynamic pricing in sponsored search auctions. *arXiv preprint arXiv:1711.10279*.

Shen, W.; Tang, P.; and Zuo, S. 2019. Automated mechanism design via neural networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 215–223. International Foundation for Autonomous Agents and Multiagent Systems.

Shen, W.; Wang, Z.; and Zuo, S. 2018. Ex-post ir dynamic auctions with cost-per-action payments. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2076–2078. International Foundation for Autonomous Agents and Multiagent Systems.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Driessche, G. V. D.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Tai, L.; Paolo, G.; and Liu, M. 2017. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *International Conference on Intelligent Robots*.

Thompson, D. R., and Leyton-Brown, K. 2013. Revenue optimization in the generalized second-price auction. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, 837–852. ACM.

Tian, F.; Li, H.; Chen, W.; Qin, T.; Chen, E.; and Liu, T.-Y. 2014. Agent behavior prediction and its generalization analysis. *arXiv preprint arXiv:1404.4960*.

Varian, H. R. 2007. Position auctions. *International Journal of industrial Organization* 25(6):1163–1178.

Wright, J. R., and Leyton-Brown, K. 2014. Level-0 meta-models for predicting human behavior in games. In *Proceedings of the fifteenth ACM conference on Economics and computation*, 857–874. ACM.

Wright, J. R., and Leyton-Brown, K. 2017. Predicting human behavior in unrepeated, simultaneous-move games. *Games and Economic Behavior* 106:16–37.

Xu, H.; Gao, B.; Yang, D.; and Liu, T.-Y. 2013. Predicting advertiser bidding behaviors in sponsored search by rationality modeling. In *Proceedings of the 22nd international conference on World Wide Web*, 1433–1444. ACM.