Optimal Online Learning in Bidding for Sponsored Search Auctions

Donghun Lee
Department of Computer Science
Princeton University
Princeton, NJ, USA
donghunl@princeton.edu

Piotr Zioło
RoomSage.com
Warsaw, Poland
piotr.ziolo@roomsage.com

Weidong Han, Warren B. Powell
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ, USA
whan, powell@princeton.edu

***Abstract*—Sponsored search advertisement auctions offer one of the most accessible automated bidding platforms to online advertisers via human-friendly web interfaces. We formulate the learning of the optimal bidding policy for sponsored search auctions as a stochastic optimization problem, and model the auctions to build a simulator based on a real world dataset focusing on the simulated sponsored search auctions to show hourly variations in auction frequency, click propensity, bidding competition, and revenue originating from advertisements. We present several bidding policies that learn from bidding results and that can be easily implemented. We also present a knowledge gradient learning policy that can guide bidding to generate samples from which the bidding policies learn. The bidding policies can be trained with a small number of samples to achieve a significant performance in advertising profit. We show that a hybrid policy that makes the optimal switching from learning the bidding policies to exploiting the learned bidding policies achieves 95% of optimal performance. Also, our result suggests that using knowledge gradient learning policy may provide robustness in guessing when to switch from learning the bidding policies to exploiting the learned bidding policies.**

I. Introduction

In 2016, it is estimated that the revenue made by search engines from placing advertisements in their search results reached $10 billion, which accounts for almost 48% of revenue from selling ad slots [1]. Naturally, these major web search engines provide human-friendly interfaces to engage in time dependent bidding for online ad auctions. For example, Google AdWords is one of such platform where advertisers can log in, place repeated bids on their advertisements, and receive statistics about the performance of their advertisement campaign on web search results from Google.

Since recently, the advertisement bidding problem has received increasing attention, and various approaches have been adopted to solve this problem. According to [2], [3], this problem can be modeled into a Knapsack problem. [2] proposes a solution based on return on investment, and characterizes the equilibrium for the bid price in a system with a given number of players with budget constraints. [3] proposes bidding strategies to solve the Knapsack problem, and provides theoretical guarantees of the proposed strategies. [4] adopts a linear programming framework that is similar to [2], and studies the convergence properties of proposed policies. In these works, the probability of winning an auction for a given bid price and the revenue of winning an auction are both assumed to be known. Our work contrasts with these work by modeling the bidding problem into a stochastic optimization problem, and allow these parameters to be unknown which can be learned over time.

[5] models the winning probability of an auction as a logistic function, based on which bidding strategies are developed. [6] forecasts the average winning bid price based on time series models, and adopts a dynamic programming approach to determine the bid price. Our problem setting is different from those in the above literature, in the sense that the bidding problem we consider is performance-based, where there incurs a cost to the advertiser if a web user clicks on the ad. Meanwhile, the problems considered in the above literature are impression-based, where there incurs a cost to the advertiser as long as the ad is displayed to the web user. Furthermore, our work contrasts with these works by adopting a Bayesian framework on random parameters instead of using point estimates, which allows as to develop learning algorithms that take into account the value of information.

This paper makes the following contributions. (1) We formulate the sponsored search advertisement problem as a stochastic optimization problem, and explicitly characterize the uncertainties faced in the real problem. (2) We explicitly define the learning policy and the bidding policy to emulate advertisers who need to gather data first to learn a bidding policy. (3) We present numerical experiment results to support that active learning policy using knowledge gradient shows better on-line performance than pure exploration policy to guide automated learning of bidding policies that can be readily used in practice.

The remainder of this paper is organized as follows. In Section II, we provide background knowledge on the problem we consider. In Section III, we provide a mathematical formulation of the problem. In Section IV, we design a simulator that mimics the process of sponsored search advertisement of a single ad campaign. In Section V, we propose algorithms that solve the bidding problem. In Section VI, we conduct several numerical experiments to compare the performances of the proposed policies against the pure exploration policy. Finally, in Section VII, we offer concluding remarks and propose directions in future research.

II. Preliminaries and Related Works

In this section, we briefly describe the Google AdWords platform, the sponsored search platform for online advertisers that serves the largest portion of the market as a single platform. We also overview the key aspects of sponsored search advertisement auctions and provide short survey on research literatures that focused on each of the aspects. Lastly, we emphasize the time dependent nature of the behavior of the online ad auctions, which was reported but has not been studied extensively from the perspective of optimizing real-time bidding policies.

A. Google AdWords Platform

When a user performs a web search using Google, the search result displayed contains not only the relevant search results but also a handful of seemingly relevant advertisements. These advertisements are placed as a result of online ad auctions, which anyone interested in placing an advertisement can access by using the Google AdWords platform. The Google AdWords platform allows users to place bids on their advertisements in advance, so as to participate in the auctions created as search queries arrive through Google search in real-time. Also, the platform computes a set of statistics from different layers of the auction process. The statistics include the number of auctions, the number of impressions, the number of clicks, the total cost incurred by clicks, and the total revenue logged due to clicks. These statistics are provided in hour-by-hour granularity, opening up possibility for human users to tune the bids hourly.

B. Key Terms to Predict

Among different statistics relevant to an online ad campaign, there are two critical quantities whose estimation has attracted considerable attention in the literature: the click-through rate (CTR) and the conversion rate (CVR) CTR estimation is performed mostly from the perspective of the auction house, as their profit models tend to be based on click counts [7] [8] [9] [10]. CTR estimation can have other latent variables, such as user preferences or ranking [11] [12].

Similarly, CVR tracks the rate of intended behavior after the advertisement is delivered. Estimating CVR has been done in the past from the perspective of major ad exchanges [13] [14] [9]. Major online ad auction houses such as Google or Yahoo are interested in CTR and CVR estimations because those are critical elements in the matching algorithms that pair up bidders to auctions. However, as conversion events occur some time after a click-through event, the data on conversions are sparser and the lags limit the ability to adjust bids in response to conversions. This scarcity of data is one of the factors that encourage most prediction and optimization attempts to predict CTR and optimize for clicks.

C. Optimization Perspectives

Optimizing in an online ad auction setting is an active field of research, where not only CTRs but also other targets of optimization has been explored. One example is to maximize the profit of the online ad auction houses by optimally matching bidders to auctions, relying on high quality estimates of CTR and CVR. Another example is to optimize matching between the demand side platform and the supply side platform of the ad exchange network, where the demand side platform serves as a proxy for advertiser and the supply side platform serves as a proxy for the auction house for online ads. It is also possible to approach the optimization problem by abstracting it into a stochastic knapsack problem with budget constraint, but this approach often assumes either true CTR or CVR is known before optimization.

We choose to focus on the fundamental purpose of advertising and optimize the profit of an individual client. Additionally, we emphasize the time dependency of the online ad auction, where bids, auction competition, and user behavior pattern changes over time. This is aligned with the basic problem an online ad client would like to solve: how to choose bids in real time for a single ad campaign hosted in a real-time bidding enabled ad auction system.

D. Time-dependent Nature of RTB

It is well known from the early stages of RTB that there are significant hourly fluctuations in the behavior of online ad auctions [15]. However, such time-dependent modeling is often not as thoroughly addressed, One way to implicitly model the time-dependency is using a number of repeated sampling to represent multiple time steps in objective functions and using exponential smoothing in estimates to give some sensitivity to time dimension in the resulting bidding strategy [8]. Another way to model time-dependency is to introduce a budget constraint. However, in the bidding problem where the budget is constrained, the time dependency is observed not from the time-dependent nature of the auction, bidding, or user behavior, but from the perspective of budget throttling as time proceeds toward a finite horizon. This problem of optimizing real-time bidding in online ad auctions deserves more explicit time-dependent modeling, because there is a natural time-dependency in the repeated online ad auctions with real-time controllable bidding strategy. The fundamental challenge is to strike the balance between learning how to maximize profits now (given an uncertainty belief) and bidding to learn the most so that better bids can be made in the future. In this paper, we design real-time learning policy that learns optimal real-time bidding policy for the online ad auction problem, exploiting the time-dependent nature of the auction.

III. Problem Statement and Modeling

In this section, we give a precise problem statement and present a fully specified model for real time bidding in repeated online ad auctions.

A. Problem Statement

We address the most fundamental problem for an advertiser who is interested in placing advertisements as sponsored search results in web search pages. The fundamental problem is how to decide the bid amount for different auctions, such

that the effect of the advertisement campaign is maximized in terms of advertisement profit. We assume that the advertisement campaign has a fixed duration of T time units, over which the performance metric is to be maximized. We also assume that we manage one auction whose characteristic changes over time. The optimization problem can be stated as:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{T-1} C\left(S_t, X^{\pi}\left(S_t\right), W_{t+1}\right)\right], \tag{1}$$

where Π is a set of policies, whose element π can be used to decide bid $x_t := X^{\pi}(S_t)$ at state S_t. We assume that the profit incurred by bid x_t is observed at time $t+1$, as the contribution function C shown in (1) is dependent on W_{t+1}, the collective randomness observable on and after time $t+1$. We denote $\hat{C}_{t+1} := C\left(S_t, X^{\pi}\left(S_t\right), W_{t+1}\right)$ to represent the observed profit, incurred by bidding x_t at time t. We set the time index t to represent each hour, such that the optimal solution of (1) is readily interpreted as the hourly optimal real-time bidding strategy to maximize the advertisement campaign profit.

B. State Variable

For all t, we define state variable $S_t = (K_t, B_t)$, whose elements to be:

- K_t, the remaining time units till the end of the advertisement campaign at time t.
- B_t is the belief state of policy $\pi \in \Pi$ at time t, whose structure depends on the policy class Π, and whose value depends on π and the sample path up to time t.

We defer specifying B_t to section V where we present different bidding policies.

Specifically, for $t = 0$, the initial state variable S_0 contains the initial information known to the policy π, before observing anything from the online ad auction simulator. The initial state variable is comprised of:

- K_0, the advertisement campaign time duration,
- B_0, the initial belief state, including the initial parameters of the bidding policy π.

We denote $\mathcal{S}$ as the state space such that all $S^t \in \mathcal{S}$.

C. Decision Variable

We plan to learn how to bid as we observe samples from the simulator as time t increases. Therefore, we define the decision variable x_t at time t as the bid amount to be placed for all online ad auctions taking place during time interval $(t, t+1]$. We discretize the space of all possible bid amounts in uniform increments into a set $\mathcal{X}$, such that all $x_t \in \mathcal{X}$. We arbitrarily set the valid bids to be $\mathcal{X} = \{0, 1, 2, \ldots, 9\}$. Bids are determined by the bidding policy X^{π}, which is a mapping from $\mathcal{S}$ to $\mathcal{X}$. We defer further discussion on how x_t is determined by B_t, as it depends on the choice of policy class Π, to section V.

D. Exogenous Information

We use W_t to denote the exogenous information provided by the auction simulator at time t. We define the set of random variables in W_t as a subset of information available to advertisers who use Google AdWords platform, described as follow:

- $\hat{N}_t^A$: Number of auctions held during time interval $(t-1, t]$.
- $\hat{N}_t^c(x)$: Number of clicks on the advertisements placed due to bidding x to auctions held during time interval $(t-1, t]$.
- $\hat{c}_t^{pc}(x)$: Cost per click, incurred by users clicking the advertisements placed due to bidding x to auctions held during time interval $(t-1, t]$.
- $\hat{r}_t^{pc}$: Revenue per click, resulted by users purchasing the advertised item after clicking the advertisements placed through auctions held during time interval $(t-1, t]$.

We defer our discussion on how to simulate these random variables to section IV.

E. Transition Function

As time proceeds from t to $t+1$, the state variable makes the transition from S_t to S_{t+1} according to the transition function S^M, such that $S^M(S_t, x_t, W_{t+1}) = S_{t+1}$, whose components make transitions as follow:

- $K_{t+1} = K_t - 1$
- $B_{t+1} = S_{\pi}^M(S_t, x_t, W_{t+1})$, where the exact update function S_{π}^M depends on the algorithm to train the chosen policy function π. This function corresponds to the learning algorithm for a policy class Π. Therefore, we defer the discussion on S_{π}^M to section V where we present different bidding policies.

F. Objective Function

Now we define the objective function, which we conceptually presented in (1), as follows:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{T-1} \left\{\hat{r}_{t+1}^{pc} - \hat{c}_{t+1}^{pc}\left(X^{\pi}(S_t)\right)\right\} \hat{N}_{t+1}^c\left(X^{\pi}(S_t)\right) \middle| S_0\right]. \tag{2}$$

This is a reworked (1), with the $C\left(S_t, X^{\pi}\left(S_t\right), W_{t+1}\right)$ term modeled as a product of the following two terms:

- $\hat{r}_{t+1}^{pc} - \hat{c}_{t+1}^{pc}\left(B^{\pi}(S_t)\right)$: the observed average profit per click by bidding $X^{\pi}(S_t)$ during time interval $(t, t+1]$, and
- $\hat{N}_{t+1}^c\left(X^{\pi}(S_t)\right)$: the observed number of clicks by bidding $X^{\pi}(S_t)$ during time interval $(t, t+1]$,

both of which are modeled as functions of bid amount $x_t = X^{\pi}(S_t)$ that is determined at time t and effective during time interval $(t, t+1]$.

IV. Online Ad Auction Simulator

In this section we present how we design the online ad auction simulator to mimic the process of sponsored search advertisement of a single ad campaign, to test different bidding policies by generating the exogenous information W_t at time t. We use a two-year-long historical dataset from a single ad campaign for one hotel to estimate parameters. The hotel providing the dataset is an average size hotel in a small city, and is representative for a large class of hotels. The hotel's ad campaign structure was prepared by a professional marketing company, again providing a very representative example of an ad campaign.

We focus on modeling the temporal patterns of online ad auctions, with the effects of different keywords and spatial attributes averaged out.

A. Number of Auctions

During time interval $(t-1,t]$, many users across the world may search for terms that trigger web search results relevant to the advertisement for which we are testing the bidding policies. The number of such search events is proportional to the actual number of auctions we face via Google AdWords system, as there may be other providers of advertisement. We use random variable N_t^A as the number of auctions in time duration $(t-1,t]$. Since random arrivals of relevant web searches trigger auctions, and users of web search engines have different activities over time, it is natural to model N_t^A to follow a time-dependent Poisson process with mean function $\lambda(h_t)$. Therefore, given t, we model the number of auctions N_t^A as follows:

$$N_t^A \sim \text{Poisson}\left(\lambda(h_t)\right), \tag{3}$$

where h_t stands for the hour-of-week of time t. It is computed as $h_t := \text{mod}\left(t + h_0, 168\right)$, where h_0 is the hour-of-week of the initial time $t = 0$. $h_t = 0$ corresponds to the first hour of Monday, and $h_t = 167$ to the last hour of Sunday. We use $\hat{N}_t^A$ to denote the actual sampled value of the random variable N_t^A. To compute $\hat{N}_t^A$, we use maximum likelihood estimators of $\lambda(h_t)$ for $h_t \in \{0, 1, \cdots, 167\}$, by fitting Poisson distributions to a two-year long historical dataset.

B. Number of Clicks

We model the conditional event of a user clicking the displayed ad given the event of the ad auction triggered by the user's web search, from the perspective of a bidder in the auction. The chain of events involved in this can be summarized into three steps. First, the bidder make a decision on how much to bid. Second, the bid was successful such that our ad is displayed in the web search result. Third, the user actually clicks on the displayed ad. We consider the probability of the above steps simultaneously, such that given a bid amount x, the probability of the bid successfully displaying the ad and the user clicking on the ad is represented in a single quantity $p^c(h(t), x)$ depending on the hour-of-week $h(t)$ and the bid amount x. We assume that the competition among the bidders and the behavior of users during time interval $(t-1,t]$ can be represented by probability $p_t^c(h(t), x)$ which is constant over $(t-1,t]$. As there are $\hat{N}_t^A$ auctions happening during time interval $(t-1,t]$, the number of clicks $N_t^c(x)$ given bid x follows a Binomial distribution as:

$$N_t^c(x) \sim \text{Binomial}\left(\hat{N}_t^A, p_t^c(h(t), x)\right). \tag{4}$$

We use $\hat{N}_t^c(x)$ to denote the sample realization of the number of clicks by bidding x at time t. We model the time-dependent behavior of bidding outcome with a logistic model, such that $p_t^c(h(t), x)$ can be computed as follows:

$$p_t^c(h(t), x) = \frac{1}{1 + e^{-\left(w^h \cdot \vec{h}(t) + \xi(x)\right)}}, \tag{5}$$

where $\vec{h}(t)$ is a 168-dimensional binary vector representing hour-of-week and w^h is the corresponding 168-dimensional weight vector for each hour-of-week. Using the additive model described above, we can model and readily interpret the level of bidding competition in auctions happening in h hour-of-week by reading off the values of w^h. The values w^h for each hour-of-week are computed by fitting (5) to the two-year historical dataset, with $\xi(x) = 1$, as the dataset was obtained with a constant bid of $x = 7$. We transform the bid x with the function $\xi(x) := \frac{\log x - 1}{\log 7 - 1}$, which ensures the following desirable properties: first, the bid $x = 7$ ensures $\xi(x) = 1$; second, as the bid $x \to 0$, the resulting $p_t^c(h(t), x) \to 0$; and third, as the bid $x \to \infty$, the resulting $p_t^c(h(t), x) \to 1$.

C. Cost per Click

We assume that the advertiser uses the pay-per-click advertisement model of Google AdWords platform, such that each click to the displayed ad gets charged. The cost per click is determined by the outcome of a Vickrey auction, where the actual cost is the bid of the runner-up. We chose to indirectly simulate the effect of bidding competition to the cost: the cost per click value we use is the smaller of the bid amount and the average cost per click in each hour-of-week from the two-year-long historical dataset. Given the bid x_t at time t, the cost per click is modeled as follows:

$$c_t^{pc}(b) = \min\left(\mu^{cpc}(h(t)), x_t\right), \tag{6}$$

where $\mu^{cpc}(h(t))$ is the historical average of cost per click in $h(t)$ hour-of-week.

D. Revenue per Click

Similar to the cost per click, we model revenue per click from a two-year-long historical dataset, by taking the average of revenue per click for each hour-of-week. Given time t, the revenue per click is modeled as follows:

$$r_t^{pc} = \mu^{rpc}(h(t)), \tag{7}$$

where $\mu^{rpc}(h(t))$ is the historical average of revenue per click in $h(t)$ hour-of-week.

V. Learning the Bidding Policies

In this section, we present several types of automated bidding policies π^{bid}. Also, we present two learning policies π^{lrn} that will guide the learning of the bidding policies. We will search for best learning policy and bidding policy that maximize the cumulative profit, which can be expressed as:

$$\max_{\pi^{lrn}} \mathbb{E}\left[\sum_{t=0}^{T-1} C\left(S_t, X^{\pi^{bid}}(S_t|\rho^{bid}), W_{t+1}\right)\middle| S_0\right], \quad (8)$$

where the updating trajectory of the bidding policy parameters depends on the learning policy π^{lrn} as $\rho^{bid} = \Theta^{\pi^{lrn}}(S_t, \rho^{lrn})$. The candidates for π^{bid} can be found in section V-A and the candidates for π^{lrn} in section V-B. The random variable C is an abbreviated term defined as:

$$\begin{aligned} &C\left(S_t, X^{\pi^{bid}}(S_t|\rho^{bid}), W_{t+1}\right) \\ &= \left\{\hat{r}^{pc}_{t+1} - \hat{c}^{pc}_{t+1}\left(X^{\pi^{bid}}(S_t|\rho^{bid})\right)\right\}\hat{N}^{c}_{t+1}\left(X^{\pi}(S_t)\right), \end{aligned} \quad (9)$$

to ensure that we are on the same objective function as shown in (2).

A. Bidding Policies

We present several different policies to decide the bid $x \in \mathcal{X}$ given a state $s \in \mathcal{S}$. For each bidding policy, we return to the modeling of beliefs B_t and the associated transition function.

1) Tabular Bidding Policy: Tabular bid policy π^{tab} is a class of bidding policy which is a form of policy function approximation [16], as it is an analytical mapping from state to action. It is defined by a table of bidding rules for each hour of week, such that a pre-set bid amount for each hour of week can be written in a bidding rule table. Naturally, this class of policy can be easily handled by human experts, whose domain expertise is utilized to set the bidding rule table. We use hour of week $h \in \mathcal{H} = \{0, 1, \cdots, 167\}$ to index the bidding rules. In general, a policy π^{tab} can be represented as:

$$\pi^{tab}\left(S_t|\rho^{tab}\right) = \rho^{tab}_{h(t)}, \quad (10)$$

where the parameter $\rho^{tab} = \left\{\rho^{tab}_h\right\}_{h\in\mathcal{H}}$ is often tuned by human experts.

Although simple in structure, when we set the tabular bidding policy with ρ^{det}_h set to be $\mu^{rpc}(h)$, the true revenue per click in hour h, it becomes the dominant bidding strategy in repeated Vickrey auctions with unlimited budget under a set of assumptions used in auction theory. We model learning this particular version of tabular bidding policy by letting the parameters ρ^{tab}_h at time t to be $\bar{r}^{pc}_{h,t}$, the estimators for true revenue per click in hour h. Since we have a discrete set of possible bids $\mathcal{X}$ as valid bid values, we present tabular bidding policy concisely as follows:

$$\pi^{tab}(S_t) = \arg\min_{x\in\mathcal{X}}\left|x - \bar{r}^{pc}_{h(t),t}\right|. \quad (11)$$

We compute the estimators $\bar{r}^{pc}_{h(t),t}$ as:

$$\bar{r}^{pc}_{i,t} = \begin{cases} \frac{1}{|\mathcal{D}_{i,t}|}\hat{r}^{pc}_t + \frac{|\mathcal{D}_{i,t}|-1}{|\mathcal{D}_{i,t}|}\bar{r}^{pc}_{i,t-1} & \text{if } i = h(t) \\ \bar{r}^{pc}_{i,t-1} & \text{otherwise} \end{cases}, \quad (12)$$

such that $\bar{r}^{pc}_{i,t}$ is the maximum likelihood estimator of $\mu^{rpc}(h)$ using data observed up to time index t, and $|\mathcal{D}_{i,t}|$ is the count of revenue per click data observed at hour i up to time index t. Therefore, the belief state B^{tab}_t can be defined as:

$$B^{tab}_t = \left\{\bar{r}^{pc}_{h,t}\right\}_{h\in\mathcal{H}}, \quad (13)$$

and the belief state changes according to (12).

This policy is equivalent to having a human expert bidder computing historical average of revenue per click for each hour of week, and updating the bidding rules in the Google AdWord platform every hour as the new data become available.

2) Linear Bidding Policy: Linear bidding policy π^{lin} is a more sophisticated bidding policy class, presented in detail in [17], which is also another policy function approximation. A linear bidding policy computes the bid as a baseline bid amount multiplied by a linear correction term. Let $\mathcal{H} = \{0, 1, \cdots, 167\}$ as an index set representing hours of week. The linear bidding policy can be represented as:

$$\pi^{lin}\left(S_t\middle|\rho^{bid}\right) = \arg\min_{x\in\mathcal{X}}\left|x - \rho^{bid}\frac{\bar{r}^{pc}_{h(t),t}}{\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\bar{r}^{pc}_{i,t}}\right|, \quad (14)$$

where $\bar{r}^{pc}_{h(t),t}$ is an estimator of revenue per click in $h(t)$ hour given observed revenue per click data up to time t. We use maximum likelihood estimator for $\bar{r}^{pc}_{h(t),t}$, which can be computed as shown in (12). ρ^{bid} is the baseline bid amount, and it is the controllable parameter for this policy, which we tuned to 1000. The value of the baseline bid amount can be chosen from, but not limited to the values in $\mathcal{X}$, as the linear correction term can be larger or smaller than 1.

This class of policy can be interpreted as bidding for a particular hour of week $h(t)$ with a tunable baseline and increasing the bid by multiplying the relative profitability of advertising in $h(t)$ hour of week compared to average profitability. The relative profitability is represented as the linear correction term of estimated revenue per click in each hour of week in (14). Also, the structure of this policy allows easy implementation in the Google AdWords platform, as the policy requires tuning the baseline bid and the adjustments for each hour of week.

3) Maximum a Posteriori Policy: The objective function in (8) can be seen as the sum of 168 separate terms corresponding to each of 168 hours in a week. To streamline notation, we assume we are at time t and hour $h(t)$. For conciseness, let $h = h(t)$. We take a Bayesian approach to estimate the profit random variable at h hour given bid x, using Gaussian distributions for the prior and the posterior distribution of each random variable the profit observed at hour h with bid x. Therefore, the belief model is to represent the h-th element of the optimization objective function shown in (2) with a Gaussian random variable with unknown mean $\mu_h(x)$ and standard deviation $\sigma_h(x)$, given a bid x. We denote $\bar{\mu}^{MAP}_{h,t}(x)$ and $\bar{\sigma}^{MAP}_{h,t}(x)$ as the estimates of $\mu_h(x)$ and $\sigma_h(x)$ at time t and bid x. The maximum a posteriori (MAP) policy π^{MAP} is a

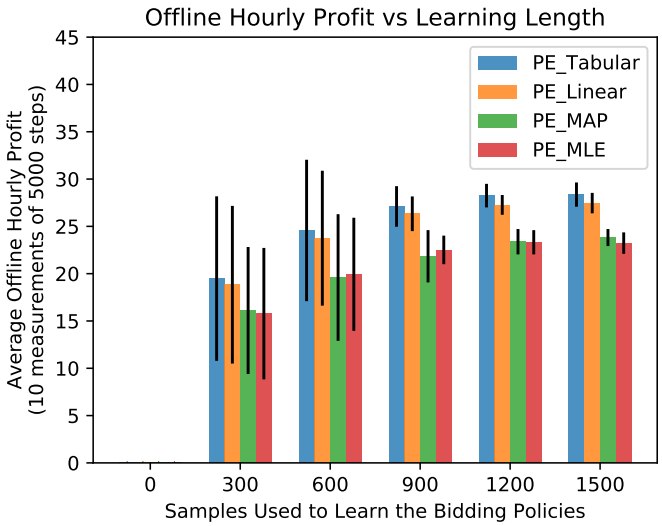

Fig. 1. Performance of bidding policies learned from a different number of samples ranging from zero to 1500 samples. The average values are computed over cumulative profit from 10 independent sample paths, where each sample path is comprised of simulating 5000 iterations. The error bars correspond to sample standard deviation. The optimal average hourly profit is 32.09.

policy that decides the bid at time t that maximizes a posteriori mean of the profit in $h(t)$ hour, which can be expressed as:

$$\pi^{MAP}(S_t|\rho^{MAP}) = \arg\max_{x\in\mathcal{X}} \bar{\mu}^{MAP}_{h(t),t}(x). \tag{15}$$

The prior mean and standard deviation parameters are arbitrarily set to $\bar{\mu}^{MAP}_{h,0}(x) = 0$ and $\bar{\sigma}^{MAP}_{h,t}(x) = 1000$.

4) Maximum Likelihood Policy: Similar to the MAP policy, we can estimate the profit random variable at h hour given bid x using maximum likelihood estimator. The maximum likelihood (MLE) policy π^{MLE} is a policy that decides the bid to maximize the maximum likelihood estimated mean of the profit given h hour of week, which can be expressed as:

$$\pi^{MLE}(S_t|\rho^{MLE}) = \arg\max_{x\in\mathcal{X}} \bar{\mu}^{MLE}_{h(t),t}(x). \tag{16}$$

We assume that the sample average of profit follows Gaussian distribution, and compute the sample mean of observed profit from bidding x in hour $h(t)$ as the estimate $\bar{\mu}^{MLE}_{h(t),t}(x)$.

B. Learning Policies

Since the bidding policies are learnable with data, we introduce the learning policy π^{lrn} to guide learning the bidding policies by interacting with the simulator. In this section, we present two types of learning policies: pure exploration (PE) and knowledge gradient (KG).

1) Pure Exploration Learning Policy: Pure exploration (PE) learning policy π^{PE} is a pure random sampling procedure. This is de facto sampling method for Monte Carlo estimation, due to the fact that statistical estimates using samples from this method is guaranteed to converge to its true values. In our problem setting, this method randomly chooses an element from $\mathcal{X}$ as bid x.

2) Knowledge Gradient Learning Policy: Knowledge gradient (KG) learning policy π^{KG} uses knowledge gradient, which quantifies the value of information in sequential decision making and makes the optimal decision on which sample to observe next [18]. The policy estimates both the mean and the variance of each bid $x \in \mathcal{X}$, and decides the bid that has maximum value of information such that the chosen bid has highest possibility make the next profit sample change our belief in which bid is optimal. Similar to the MAP policy, pi^{KG} assumes that the advertisement profits follow Gaussian distributions with unknown mean $\mu_h(x)$ and standard deviation $\sigma_h(x)$ that depend on hour of week h. We denote $\bar{\mu}^{KG}_{h(t),t}(x)$ and $\bar{\sigma}^{KG}_{h(t),t}(x)$ as the posterior mean and standard deviation parameter estimates of $\mu_{h(t)}(x)$ and $\sigma_{h(t)}(x)$ using data observed from the simulator up to time t. Then, the belief state B^{KG}_t for knowledge gradient policy is defined as:

$$B^{KG}_t = \left\{\bar{\mu}^{KG}_{h(t),t}(x), \bar{\sigma}^{KG}_{h(t),t}(x)\right\}_{(h(t),x)\in\mathcal{H}\times\mathcal{X}} \tag{17}$$

The knowledge gradient policy $\pi^{KG}(S_t)$ determines the next bid as follows:

$$\pi^{KG}(S_t) = \arg\max_{x\in\mathcal{X}} \tilde{\sigma}_{h(t),t}(x)\left(\zeta^t_x \Phi(\zeta^t_x) - \phi(\zeta^t_x)\right), \tag{18}$$

where Φ and ϕ are cdf and pdf of standard normal distribution, and ζ^t_x is defined as:

$$\zeta^t_x = -\left|\frac{\bar{\mu}^{KG}_{h(t),t}(x) - \max_{x'\neq x}\bar{\mu}^{KG}_{h(t),t}(x')}{\tilde{\sigma}^t_{h(t),t}(x)}\right|, \tag{19}$$

and $\tilde{\sigma}_{i,t}(x)$ for $i \in \mathcal{H}$ is computed as:

$$\tilde{\sigma}_{i,t}(x) = \begin{cases} \sqrt{\left(\bar{\sigma}^{KG}_{i,t}(x)\right)^2 - \left(\bar{\sigma}^{KG}_{i,t}(x)\right)^2} & \text{if } i = h(t) \\ \tilde{\sigma}_{i,t-1}(x) & \text{otherwise} \end{cases}. \tag{20}$$

VI. Empirical Verification

In this section, we present the results of optimizing (8) for different bidding policies and learning policies using the online ad auction simulator presented in section IV. We compare the performance of different bidding policies and learning policies. Considering the practical issue of scarce number of samples from ad auction time series data, we report the empirical behavior of policies and learning algorithms over a small number of samples from our simulator. As we model one time step as one hour, we note that 168 iterations correspond to a week, 720 to a month (of 30 days), and 2184 to a quarter of a year.

A. Offline Performance of Bidding Policies

We first test how fast each bidding policy learn, by comparing the performance of learned bidding policies measured by offline average hourly profit. We use pure exploration (PE) learning policy to provide unbiased samples from the auctions simulator, and each bidding policies learned from a different number of samples ranging from 0 to 1500 samples. We perform 10 repeats of learning the bidding policies, and

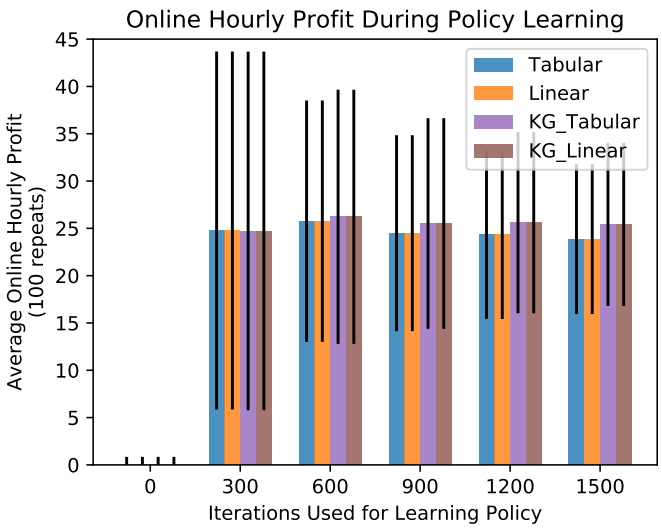

Fig. 2. Performance as the operating profit measured while learning bidding policies, over different number of learning samples used. The average values are computed over cumulative profit at six different points of 100 independent sample paths, where each sample path is comprised of simulating 1500 iterations. The error bars correspond to sample standard deviation. The optimal average hourly profit is 32.09.

the performance of each learned bidding policy is measured in auction simulations of length 5000 iterations. The result, shown in Figure 1, clearly indicates that tabular bidding policy and linear bidding policy has better sample efficiency than the other bidding policies. The bidding policies are not given a priori information about the auctions, which is shown as zero performance with zero learning samples. After 1500 random samples, which is roughly equivalent to data collected in 10 weeks by random bidding according to PE learning policy, the learned bidding policies achieve 80-90% of optimal hourly profit of 32.09.

B. Online Performance of Learning Policies

In addition to the offline performance of learned bidding policies after 1500 samples from auctions show the average hourly profit after learning, we present the online performance of learning policies, measured by average hourly profit from generating the 1500 learning samples.

As described in section V-B, we have two learning policies to accumulate learning data: pure exploration (PE) policy and knowledge gradient (KG) policy. We test the learning policies for tabular bidding policy and linear bidding policy, which excelled the other two policies in offline performance. The result, shown in Figure 2, suggests that both learning policies show insignificant difference in online performance due to large variance among sample paths shown in wide error bars. Also, as the offline performance of learned bidding policies excels the online performance of learning policies roughly after 900 samples for π^{tab} and π^{lin}, this result suggests finding the best number of iterations to switch from learning the bidding policies to exploiting the learned policies.

C. Balancing Exploration and Exploitation

The experiment results shown in previous sections suggest that there may be the best switching moment from using learning policies to using learned bidding policies. This is an exploration-exploration problem for learning how to bid for sponsored search auctions. This gives a twist to the problem such that we are now optimizing a slightly different objective function:

$$\max_{U\in[0,T]} \left\{ \mathbb{E}\left[\sum_{t=0}^{U-1} C\left(X_t^{\pi^{lrn}}\right) \middle| S_0 \right] + \mathbb{E}\left[\sum_{t=U}^{T-1} C\left(X_t^{\pi^{bid}}\right) \middle| S_U \right] \right\}, \tag{21}$$

where U is the tunable switching parameter, and $C\left(X_t^{\pi}\right) = C\left(S_t, X^{\pi}(S_t|\rho), W_{t+1}\right)$ whose full definition is given in (9).

Using the simulator, we fix the advertisement campaign horizon $T = 5000$ (corresponding to roughly half a year), and vary the length of learning period U such that $0 \leq U \leq T$ to find the optimal value of U that maximizes the objective function (21). We search for $U \in \{0, 250, 500, \cdots, 5000\}$ to represent the fact that it is possible to switch from learning to exploiting at any moment throughout the ad campaign duration.

The result, shown in Figure 3, suggests that the best switching parameter U may be found around 1000 or 1500 depending on which learning policy is used, for π^{tab} the bidding policy which shows the best online performance among bidding policies. It is notable that KG learning policy would allow a wider margin of error in deciding U than PE learning policy for both π^{tab} and π^{lin}, when required to pick a U without tuning. This is shown in Figure 3 as a wider hump around $U = 1500$ with slower decay of hourly profit for both $KG\_Tabular$ and KG_Linear, compared to a faster decaying hump for $PE\_Tabular$ and PE_Linear around $U = 1000$. Also, with properly chosen U, the hybrid application of first using learning policy and then using the learned bidding policy results in online performance approaching nearly 95% of the optimal hourly profit of 32.09.

VII. Conclusion and Future Works

In this paper, we formulate the sponsored search advertisement problem as a stochastic optimization problem and construct an ad auction simulation model using a real world dataset. We implement a simulator following the model, and test the offline and online performance of several different automated bidding policies that learn from participating in auctions by bidding and observing profits. Moreover, we compare learning policies, one unbiased, and the other based on knowledge gradient, to generate data to learn the bidding policies. We demonstrate that it is possible to search for optimal switching time from learning to exploiting to achieve 95% of optimal profit from the simulated auctions. Also, we conjecture that using knowledge gradient learning policy may provide greater robustness in choosing the optimal switching

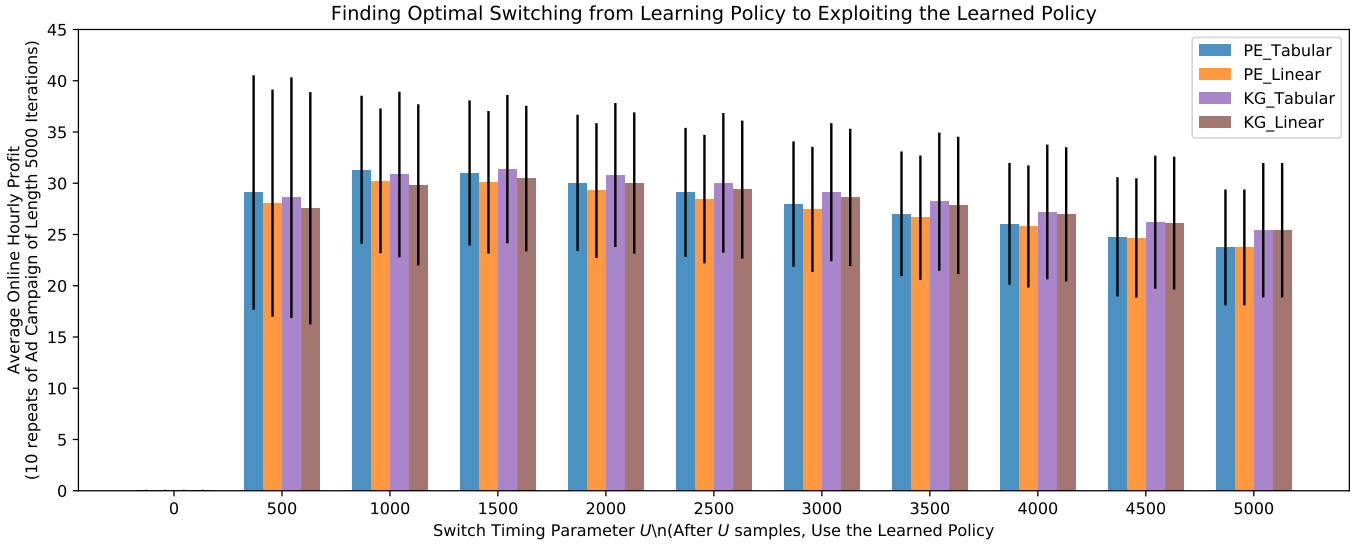

Fig. 3. Online hourly profit measured over 5000 iterations, where first U samples are used to learn the bidding policy and for the remaining iterations the learned bidding policy decides the bids to maximize the online profit. The average values are computed on online cumulative profits divided by the total number of iterations, for each of U values shown as the horizontal axis, measured in 10 independent sample paths, where each sample path is comprised of simulating 5000 iterations. The error bars correspond to sample standard deviation. The optimal average hourly profit is 32.09.

time compared to pure exploration learning policy. In future, we plan to extend the knowledge gradient learning policy such that it would find the optimal switching threshold on-the-fly. Also, we expect to enhance the sponsored search auction simulator to represent additional diversity in real world such as geographical effect and user-specific variances in order to emulate the complex nature of the auctions more accurately.

Acknowledgment

We gratefully acknowledge the support of RoomSage.com by Wizard Forms Ltd., with the historical bidding dataset and funding from the European Union.

References

[1] PricewaterhouseCoopers. (2017, April) IAB internet advertising revenue report, 2016 full year results. Interactive Advertising Bureau.

[2] C. Borgs, J. Chayes, N. Immorlica, K. Jain, O. Etesami, and M. Mahdian, "Dynamics of bid optimization in online advertisement auctions," in *Proceedings of the 16th International Conference on World Wide Web*. ACM, 2007, pp. 531–540.

[3] Y. Zhou, D. Chakrabarty, and R. Lukose, *Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems*, 2008, pp. 566–576.

[4] N. Chaitanya and Y. Narahari, "Optimal equilibrium bidding strategies for budget constrained bidders in sponsored search auctions," *Operational Research*, vol. 12, no. 3, pp. 317–343, 2012.

[5] X. Li and D. Guan, *Programmatic Buying Bidding Strategies with Win Rate and Winning Price Estimation in Real Time Mobile Advertising*. Cham: Springer International Publishing, 2014, pp. 447–460.

[6] S. Adikari and K. Dutta, *Real Time Bidding in Online Digital Advertisement*. Springer International Publishing, 2015, pp. 19–38.

[7] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: Estimating the click-through rate for new ads," in *Proceedings of the 16th International World Wide Web Conference(WWW-2007)*, January 2007.

[8] T. Graepel, J. Quionero Candela, T. Borchert, and R. Herbrich, "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsofts bing search engine," in *Proceedings of the 27th International Conference on Machine Learning ICML 2010, Invited Applications Track (unreviewed, to appear)*, June 2010.

[9] O. Chapelle, E. Manavoglu, and R. Rosales, "Simple and scalable response prediction for display advertising," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 4, pp. 61:1–61:34, Dec. 2014.

[10] Q. Liu, F. Yu, S. Wu, and L. Wang, "A convolutional click prediction model," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '15. New York, NY, USA: ACM, 2015, pp. 1743–1746.

[11] S. Shen, B. Hu, W. Chen, and Q. Yang, "Personalized click model through collaborative filtering," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, ser. WSDM '12. New York, NY, USA: ACM, 2012, pp. 323–332.

[12] Y. Tagami, S. Ono, K. Yamamoto, K. Tsukamoto, and A. Tajima, "Ctr prediction for contextual advertising: Learning-to-rank approach," in *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, ser. ADKDD '13. New York, NY, USA: ACM, 2013, pp. 4:1–4:8.

[13] K.-c. Lee, B. Orten, A. Dasdan, and W. Li, "Estimating conversion rate in display advertising from past performance data," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 768–776.

[14] R. Rosales, H. Cheng, and E. Manavoglu, "Post-click conversion modeling and analysis for non-guaranteed delivery display advertising," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, ser. WSDM '12. New York, NY, USA: ACM, 2012, pp. 293–302.

[15] S. Yuan, J. Wang, and X. Zhao, "Real-time bidding for online advertising: Measurement and analysis," in *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, ser. ADKDD '13. New York, NY, USA: ACM, 2013, pp. 3:1–3:8.

[16] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality.*, 2nd ed. John Wiley & Sons, Inc., 2011.

[17] C. Perlich, B. Dalessandro, R. Hook, O. Stitelman, T. Raeder, and F. Provost, "Bid optimizing and inventory scoring in targeted online advertising," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 804–812.

[18] W. B. Powell and I. O. Ryzhov, *Optimal Learning*. John Wiley & Sons, Inc., 2012.