Efficient Regret Bounds for Online Bid Optimisation in Budget-Limited Sponsored Search Auctions

Long Tran-Thanh[1], Lampros Stavrogiannis[1], Victor Naroditskiy[1]
Valentin Robu[1], Nicholas R Jennings[1] and Peter Key[2]
1: University of Southampton, UK
{ltt08r, ls8g09, vn, vr2, nrj}@ecs.soton.ac.uk
2: Microsoft Research Cambridge, UK
peter.key@microsoft.com

Abstract

We study the problem of an advertising agent who needs to intelligently distribute her budget across a sequence of online keyword bidding auctions. We assume the closing price of each auction is governed by the same unknown distribution, and study the problem of making provably optimal bidding decisions. Learning the distribution is done under censored observations, i.e. the closing price of an auction is revealed only if the bid we place is above it. We consider three algorithms, namely $\varepsilon-$First, Greedy Product-Limit (GPL) and LuekerLearn, respectively, and we show that these algorithms provably achieve Hannan-consistency. In particular, we show that the regret bound of $\varepsilon-$First is at most $O(T^{\frac{2}{3}})$ with high probability. For the other two algorithms, we first prove that, by using a censored data distribution estimator proposed by Zeng [19], the empirical distribution of the closing market price converges in probability to its true distribution with a $O(\frac{1}{\sqrt{t}})$ rate, where t is the number of updates. Based on this result, we prove that both GPL and LuekerLearn achieve $O(\sqrt{T})$ regret bound with high probability. This in fact provides an affirmative answer to the research question raised in [1]. We also evaluate the abovementioned algorithms using real bidding data, and show that although GPL achieves the best performance on average (up to 90% of the optimal solution), its long running time may limit its suitability in practice. By contrast, LuekerLearn and $\varepsilon-$First proposed in this paper achieve up to 85% of the optimal, but with an exponential reduction in computational complexity (a saving up to 95%, compared to GPL).

1 Introduction

Sponsored search is the most significant example of monetisation of Internet activities. This multi-billion dollar industry poses many challenging research problems for both advertisers and search engines. One of the most well-studied, but nonetheless still open, problems is the optimisation of marketing campaigns for an advertiser, or an autonomous agent acting on her behalf[1], with a fixed budget. This fundamental problem has been studied in a number of stylised models, yet many of the questions arising in real sponsored search auctions remain unanswered. In this paper, we focus on one such question—bidding when prices are not known but must be learnt to choose the right bidding strategy.

In this work, we follow a stochastic market price model that was used in [1, 7]. In particular, we take the point of view of an advertising agent with a specified budget for a given time horizon, who wants to find a bidding strategy that maximises the number of clicks. We consider a model with a single keyword and a single slot. Each time a user searches for the keyword, an auction is run to decide which of the interested agents is assigned the ad slot on the search results page. The winner is the agent with the highest bid who pays the *market price* which is determined by the second highest bid: i.e., the slot is sold in the style of a second-price auction. In practice, other factors affecting allocation of the slot include randomisation used by the search engine and advertiser/keyword-specific "quality scores" that adjust advertisers' bids. Given these factors and the lack of information about bids and strategies of the other advertisers, an advertising agent cannot easily take into account her own effect on the market price, and so instead views the price as a random variable.

The key challenge of this stochastic model is that the distribution of the market price is not known in ad-

[1]We will interchangeably use the terms agent and advertiser within this paper.

vance. Thus, to select the right bidding strategy, the agent needs to learn that distribution. This learning problem is further complicated by "censored observations" [1, 7]: the agent observes the market price only when she wins the auction; otherwise she just learns that the market price is above her bid. Although existing methods, designed for budget-limited online optimisation, can provably achieve asymptotically optimal performance for the case with no censorship [2, 18], they fail within the settings studied here. In particular, due to the censored observations, these methods cannot reproduce efficient estimation of the distribution of the market price, as they use conventional empirical estimation techniques [9, 16].

To combat this, Amin *et al.* (2012) proposed Greedy Product-Limit (GPL) and LuekerLearn, two methods that use the Kaplan–Meier estimator [9], designed for estimating the distribution of censored data, and achieve good performance in experiments with real bidding data. However, no theoretical performance analysis has been provided. Against this background, this paper addresses this gap by providing theoretical justification for these algorithms and a novel one that we develop for this setting. Our results prove asymptotic optimality of the algorithms, guaranteeing good performance as the number of auctions increases. We first look at $\varepsilon-$First, an algorithm inspired by a class of methods designed for multi-armed bandits [5, 18], tailored to our settings. In particular, this algorithm uses the first ε fraction of the auctions to estimate the market price distribution. Based on this estimate, it then solves a Markov decision process (MDP) in order to determine the optimal bidding policy. We prove that this algorithm achieves Hannan–consistency (i.e., sub–linear regret bound). Put differently, we show that the regret (i.e., the difference between the performance of a particular algorithm and that of an optimal solution) of the algorithm is at most $O(T^{\frac{2}{3}})$ with high probability, where T is the number of auctions. Note that the Hannan–consistency property (i.e., the sub-linear $O(T^{\frac{2}{3}})$ regret bound) guarantees that the average regret (i.e., the total regret divided by the number of auctions) converges to 0 as the number of auctions is increased, and thus, the bidding behaviour of a Hannan–consistent algorithm becomes more similar to that of the optimal solution (due to the decreasing performance gap defined by the average regret).

In addition to $\varepsilon-$First, we also provide an affirmative answer to the conjectures posed in [1]. That is, we show that, by replacing the Kaplan–Meier estimator with a novel censored data distribution estimator proposed by Zeng [19], GPL and LuekerLearn, the algorithms studied by Amin *et al.*, do indeed achieve sub–linear regret bounds, and thus, are also Hannan–consistent. In particular, we show that, by using Zeng's estimator, the empirical distribution of the closing market price converges in probability to its true distribution with a $O(\frac{1}{\sqrt{t}})$ rate, where t is the number of updates. Relying on this result, we prove that GPL achieves $O(\sqrt{T})$ regret bound with high probability. On the other hand, LuekerLearn achieves $O(\sqrt{T}+\ln T)$ regret bound, also with high probability. Given this, our work extends the state of the art as follows:

- We provide a theoretical regret analysis for $\varepsilon-$First, GPL and LuekerLearn, and we show that they achieve Hannan–consistency.
- We compare the performance of each algorithm through extensive empirical evaluations, using real bidding data from Microsoft adCenter. In particular, we demonstrate that, although GPL typically outperforms the other algorithms, it requires significantly higher computational complexity, which could limit its suitability in practice. On the other hand, both $\varepsilon-$First and LuekerLearn can achieve performance close to that of GPL (typically within 10%), but with a much lower computational cost (with up to 50 times speed-up in computation time).

The remainder of the paper is organised as follows. In the next section we review related work. The model we study is presented in Section 3. We review existing and new algorithms for learning and bidding in Section 4. Our main contribution — theoretical guarantees — are derived in Section 5. Numerical evaluation using real-world data sets is presented in Section 6, and Section 7 concludes.

2 Related Work

Bid optimisation in sponsored search auctions is a topic of considerable research in the autonomous agents community [4, 8, 10, 12, 14]. One of the first papers on the topic offers heuristic algorithms for prediction and bidding that were shown to work in practice [10]. Moreover, Feldman *et al.* [6] prove that simple randomised strategies for optimising a budget across multiple keywords achieve good performance. In that work, cost per click and number of clicks for each bid are known to the bidder. Berg *et al.* [3] compare bidding algorithms such as equating return-on-investment (ROI) and knapsack-based solutions based on the predictions they require (e.g., number of clicks and cost per click) and evaluate them in a simulated bidding environment of the Trading Agent Competition in Ad Auctions [8]. Unlike all the above papers, our focus is on bidding with online *learning*—in order to make bidding decisions, we need to learn the distribution of the market price.

The two papers closest to our work that combine learning and bidding in ad auctions are [1] and [20]. In par-

ticular, our work can be seen as a continuation of the research started by Amin *et al.* [1] who compared various algorithms for prediction and bidding. We adopt the same model, but focus on theoretical guarantees of the algorithms considered in [1] as well as that of our proposed $\varepsilon-$First algorithm. Zhou and Naroditskiy [20] address the keyword bidding problem when multiple slots are available, but do not provide theoretical guarantees for their proposed algorithm.

Furthermore, two very recent related works are [18] and [2]. Both works propose general frameworks for studying multi-armed bandit problems with supply (or budget) constraints. Although bidding in repeated auctions is a problem that can be modeled in these frameworks, they do not address the one-sided censored observations issue, which is the main challenge addressed here. It is worth noting that we can still apply these models to our settings by combining them with the censored data solutions described in Section 4. However, since they are designed for more generic problems, they do not exploit the domain–specific features of our problem, and thus, they provide weaker performance guarantees. Nevertheless, they may form a strong basis for our future work.

Finally, it is worth to note that our problem can also be formalised as a Markov decision process (MDP) (see Section 3 for more details), and thus, it shows similarities to the domain of reinforcement learning [15, 17]. However, as existing RL methods do not take into account censored data, it is not trivial how to incorporate them into our settings. Given this, we ignore the large literature of RL, as we argue that they are out of scope of our paper. Nevertheless, a possible future work would be to find an efficient way to combine RL techniques with censored data estimation.

3 Model Description

Our model consists of a sequence of T single slot second–price auctions, where the bidder (or agent) has to repeatedly place her bid in order to win a single keyword at each time step $t \in \{1, \dots, T\}$. We refer to T auctions as a bidding *period* and use B to denote the budget for the period. That is, the total cost spent on the auctions cannot exceed this budget. At each time step t, we assume that the market price x_t of the keyword is an independent and identically distributed (i.i.d.) random variable drawn from an unknown, but fixed, distribution with probability distribution function p. We assume that p has a finite support $[0, C]$ for some sufficiently large $C > 0$. This assumption is reasonable, as the market price is typically less than a couple of dollars. In our model, if a particular bid of the agent at time step t is higher than x_t, the agent wins the auction, and the budget is decreased by x_t. Otherwise, the agent does not win, and the budget remains the same. More formally, let b_t and B_t denote the agent's bid, and the residual budget (i.e., the remaining budget) at time step t, respectively. Note that $B_1 = B$. Given this, we have

$$B_{t+1} = B_t - x_t$$

if $b_t \geq x_t$, and

$$B_{t+1} = B_t$$

otherwise. Note that the agent cannot place a bid that is higher than the current residual budget. That is, $b_t \leq B_t$ for each time step t. We assume that both b_t and x_t are discrete values chosen/drawn from $\mathbb{Z}^+$. This assumption is reasonable, as the bids and market prices can be regarded as multiplications of the smallest unit of currency allowed for bidding.

Now, our goal is to find a bidding policy that maximises the number of wins over the time interval $\{1, \dots, T\}$. It is worth to note that if $B \geq CT$, we can achieve the optimal solution by repeatedly bidding with C. In particular, since bidding C always guarantees winning, if our budget is larger than CT, we can always win at each time step. Given this, we now only focus on the nontrivial case, and thus, we from hereafter assume that

$$B < CT \tag{1}$$

Given this condition, our problem can be formalised as follows. Let A denote a bidding policy that places bid $b^A(B_t, t)$ at each time step t , where B_t is the residual budget at that time step. In addition, let $G^A(B, T)$ denote the expected total number of wins of policy A with respect to total budget B and time limit T:

$$G^A(B, T) = \mathbb{E}\left[\sum_{t=1}^{T} I\{b^A(B_t, t) \geq x_t\}\right], \tag{2}$$

where $I\{\cdot\}$ is the indicator function. Note that $b^A(B_t, t) \leq B_t$ and

$$B_{t+1} = \begin{cases} B_t - x_t, & \text{if} \quad b^A(B_t, t) \geq x_t \\ B_t, & \text{otherwise.} \end{cases}$$

We aim to find an optimal policy

$$A^* = \arg\max_A G^A(B, T)$$

that maximises the expected total number of wins. For the sake of simplicity, we denote the expected performance of A^* with $G^*(B, T)$. It is known that if we have exact information about the distribution function p, we can calculate A^* using a Markov decision process (MDP) formulation [1, 15] . In particular, let $F(b) = P(X > b)$ denote the survival function of the market price[2] (i.e., the probability that the market

[2]The problem of estimating the distribution of censored data first appeared in the survival analysis literature [9, 13]. Hence the name of the survival function.

price is higher than bid b). Suppose that the optimal policy A^* chooses bid $b^*(B',t)$ if the budget is B' at time step t. It can be shown that A^* has to satisfy the following set of Bellman equations [15]:

$$\begin{aligned} b^*(B',t) &= \arg\max_{b(B',t)} \left\{ \sum_{\sigma=1}^{b(B',t)} p(\sigma) \left[1 + G^*(B'-\sigma, T-t)\right] \right. \\ &+ \left. F(b(B',t)) G^*(B', T-t) \right\} \end{aligned}$$

$$\begin{aligned} G^*(B', T-t+1) &= \sum_{\sigma=1}^{b^*(B',t)} p(\sigma) \left[1 + G^*(B'-\sigma, T-t)\right] \\ &+ F(b^*(B',t)) G^*(B', T-t) \end{aligned}$$

for each $t \in \{1, \ldots, T\}$ and $0 \leq B' \leq B$. That is, $b^*(B',t)$ denotes the optimal bid (i.e., the one that maximises the expected number of future wins) at time step t and budget B', while the second equation implies that the optimal number of wins at time step t and budget B' can be achieved by taking the optimal bid and continuing with the optimal policy A^* (for more details, see e.g. [1, 15]). Note that $G^*(B',0) = 0$ for any $0 \leq B' \leq B$. Given this, we can recursively solve the Bellman equations given above, and thus, determine the optimal bid for each time step t in order to calculate the optimal solution $G^*(B,T)$. Hereafter we may refer to A^* as the optimal stochastic solution, as opposed to the deterministic approach, that additionally has full information about the sequence of market prices x_t, which A^* typically does not have (see Section 4.3 for more details).

Since p is unknown for us, A^* cannot be determined in an exact manner. This implies that A^* represents a theoretical optimum value, which is unachievable in general. Nevertheless, for any algorithm A, we can define the regret for A as the difference between the total number of wins of A and that of the theoretical optimum A^*. More precisely, letting R^A denote the regret, we have

$$R^A(B,T) = G^*(B,T) - G^A(B,T)$$

Thus, our objective is to derive algorithms for learning p and bidding that minimise this regret.

4 Algorithms

Given the problem definition, we now turn to the description of the algorithms that we study within this paper. In particular, we investigate three algorithms: (i) $\varepsilon-$First, (ii) GPL and (iii) LuekerLearn. These algorithms are described in the next sections.

4.1 The $\varepsilon-$First Algorithm

Algorithm 1 The $\varepsilon-$First Algorithm

```
Inputs: T > 0, B > 0, 0 < ε < 1;
Exploration phase:
for t = 1 → εT do
    randomly choose bid b_t from uniform distribution over [1, B/(εT)];
    observe o_t = min{x_t, b_t};
end for
Exploitation phase:
use Suzukawa's estimator to calculate p̂;
solve the Bellman equations given in Equation 4;
for t = εT → T do
    place the bid b^+(B_t, t) accordingly to the solution of the Bellman equations;
end for
```

As mentioned earlier, the key challenge of finding an optimal solution for the budget–limited auction problem is that we do not know the distribution function p of the market price in advance. Given this, we need to learn (or estimate) this distribution from the observed sequence of market prices $x_1, x_2, \ldots, x_T$. This naturally lends itself to the idea of $\varepsilon-$First, which first estimates the distribution of the market price and then optimises the bidding policy. In particular, it uses an ε fraction of the total number of auctions T within a period to estimate the market price distribution function p. Following this, in the rest of $(1-\varepsilon)\,T$ auctions, we solve the budget–limited auction problem with the estimated market price distribution function $\hat{p}$ learnt from the learning phase. Hereafter we refer to the former phase as *exploration*, while to the latter as *exploitation*, respectively. In what follows, we describe these phases in more detail (the pseudo code is depicted in Algorithm 1).

We start with the description of the exploration phase. Within this phase, our goal is to accurately estimate the market price distribution. To do so, we can use the first ε proportion of the total auctions T. Now, recall that we can only observe x_t when it is not higher than the chosen bid b_t. That is, the sequence of x_t is (right) censored by the sequence of b_t. In particular, at each time step, we can only observe the value of $o_t = \min\{x_t, b_t\}$. This, indeed, makes the estimation of p a challenging problem. Note that [1] used the product–limit, or Kaplan–Meier (KM), estimator to address this challenge [9]. However, it is well known that the KM estimator has a negative bias [13]. To overcome this issue, we consider a modification of the KM estimator, an estimation technique proposed by [16], for estimating functionals of the distribution p. This method is proven to be unbiased, and thus, we can use McDiarmid's inequality to guarantee the $O\left(t^{-1}\right)$ convergence rate of the $\hat{p}_t$ estimate. This convergence rate provides the basis for the performance analysis of $\varepsilon-$First (see Section 5 for more details).

Suzukawa's method can be adopted to the estimation

of the market price distribution as follows. It relies on the assumption that we know the distribution from which the bids b_t are drawn. Let S denote the survival function of this bid distribution, and $o_t = \min\{x_t, b_t\}$ denote the observed value at t. Let $\varphi_b(x)$ be a function defined as

$$\begin{aligned} \varphi_b(x) &= 1 \quad \text{if} \quad x \leq b \\ \varphi_b(x) &= 0 \quad \text{otherwise.} \end{aligned}$$

In addition, let $\delta_t = I\{x_t \leq b_t\}$ denote the indicator function whether the market price does not exceed the bid at time step t. Given this, Suzukawa's estimation for the market price's cumulative probability function P is formalised as:

$$\hat{P}_t(b) = \frac{1}{t}\sum_{i=1}^{t}\frac{\delta_i \varphi_b(o_i)}{S^-(o_i)} \tag{3}$$

where $S^-(o_i) = \lim_{x>0, x\to 0} S(o_i - x)$ and $\hat{P}_t(b)$ is the estimate of $P(b)$ after t observations. Using techniques similar to those from [16], we can easily derive that $\hat{P}_t(b)$ is indeed an unbiased estimator of $P(b)$.

Based on this, $\varepsilon-$First places the bids within the exploration phase as follows. For each $t \leq \varepsilon T$, $\varepsilon-$First uniformly chooses a bid b_t from $\left[1, \frac{B}{\varepsilon T}\right]$ (Algorithm 1, lines $4-5$). This guarantees that the total cost spent within the exploration phase will not exceed the total budget B. When the exploration ends, let $\hat{p}$ and $\hat{F}$ denote Suzukawa's KM estimation of the market price distribution function p, and the survival function, respectively (line 8). Next, we will describe how $\varepsilon-$First uses these estimates to tackle the budget–limited auction problem.

We now turn to the description of the exploitation phase. Let $B_{\varepsilon T}$ denote the residual budget after the exploration phase ends. In order to determine the bids at each time step, $\varepsilon-$First solves the following Bellman equations:

$$\begin{aligned} b^+(B', t) &= \arg\max_{b(B',t)} \left\{ \sum_{\sigma=1}^{b(B',t)} \hat{p}(\sigma)\left[1 + G^+(B'-\sigma, T-t)\right] \right. \\ &+ \left. \hat{F}(b(B',t)) G^+(B', T-t) \right\} \\ G^+(B', T-t+1) &= \sum_{\sigma=1}^{b^+(B',t)} \hat{p}(\sigma)\left[1 + G^+(B'-\sigma, T-t)\right] \\ &+ \hat{F}(b^+(B',t)) G^+(B', T-t) \end{aligned} \tag{4}$$

for each $\varepsilon T \leq t \leq T$ and $0 \leq B' \leq B_{\varepsilon T}$, where $b^+(B', t)$ is the chosen bid of $\varepsilon-$First at time step t and budget B'. Recall that $G^+(B', 0) = 0$ for any $0 \leq B' \leq B$. These together allow us to (recursively) evaluate each value of $b^+(B', t)$, and thus, the bidding policy within the exploitation phase of $\varepsilon-$First (Algorithm 1, lines $9-12$).

Algorithm 2 The GPL Algorithm

1: **Inputs:** $T > 0$, $B > 0$, $\hat{p}_1$ is uniform;
2: **for** $t = 1 \to T$ **do**
3: solve the Bellman equations given in Equation 5 for $\hat{p}_t$;
4: place a bid $b^+(B_t, t)$ according to the solution of the Bellman equations;
5: use Zeng's estimator to update $\hat{p}_{t+1}$;
6: **end for**

The intuition behind $\varepsilon-$First is that by properly setting the value of ε, we can quickly estimate the distribution of the market price with sufficient accuracy. Thus, the solution of the Bellman equations within the exploitation phase is close to the optimal solution, resulting in a good overall bidding performance (see Section 5 for more details).

4.2 The GPL Algorithm

The GPL algorithm, introduced by [1], can be described as follows. For each time step t, it uses an MDP model to determine the current optimal policy, given the current estimate $\hat{p}_t$ of the market price distribution function p. That is, it solves a set of Bellman equations, similar to the exploitation phase of $\varepsilon-$First, but with a different $\hat{p}_t$ at each time step. According to this optimal policy, it then chooses the next bid, and observes the censored value o_t. Based on this observation, GPL uses a novel censored data distribution estimator, proposed by [19], to update the estimation of the market price distribution function, $\hat{p}_{t+1}$, for the next time step. Note that here we replace the KM estimator, which is used in [1], with Zeng's method (for a brief description of Zeng's method and further explanations, see Section 5). The above mentioned steps are repeated until $t = T$ (see Algorithm 2 for the pseudo code). More formally, suppose that the residual budget at time step t is B_t. In addition, let $\hat{F}_t$ denote the estimate of the survival function at t. GPL solves the following equations:

$$\begin{aligned} b^+(B', \tau) &= \arg\max_{b(B',\tau)} \left\{ \sum_{\sigma=1}^{b(B',\tau)} \hat{p}_t(\sigma)\left[1 + G^+(B'-\sigma, T-\tau)\right] \right. \\ &+ \left. \hat{F}_t(b(B',t)) G^+(B', T-\tau) \right\} \\ G^+(B', T-\tau) &= \sum_{\sigma=1}^{b^+(B',\tau)} \hat{p}_t(\sigma)\left[1 + G^+(B'-\sigma, T-\tau-1)\right] \\ &+ \hat{F}_t(b^+(B',\tau)) G^+(B', T-\tau-1) \end{aligned} \tag{5}$$

where $t \leq \tau \leq T-1$ and $0 \leq B' \leq B_t$. In addition, we have $G^+(B', 0) = 0$ for all $0 \leq B' \leq B_t$. For the sake of simplicity, we set $\hat{p}_1$ to be a uniform distribution in $(0, B]$. Given the solutions, GPL then places bid $b^+(B_t, t)$ at each time step t (Algorithm 2, lines $2-6$).

4.3 The LuekerLearn Algorithm

Similar to GPL, this algorithm was also described in [1], and is based on the algorithm proposed by Lueker for the online stochastic knapsack problem [11]. In particular, within the online stochastic knapsack problem, an item with profit r_t and weight x_t arrives into the system at each time step t such that the pair $\{r_t, x_t\}$ is drawn from a *fixed* and *known* joint distribution. At each time step, we have to decide whether to put the arrived item into a knapsack with the total capacity B such that the total weight of the chosen items cannot exceed this capacity. Our goal is to maximise the total profit of the chosen items. It is easy to see that within our settings, if the market price distribution p is known in advance, the budget–limited auction problem can be reduced to the online stochastic knapsack problem by setting $r_t = 1$ for each t. Given this, Lueker's algorithm, originally designed for the online stochastic knapsack problem, can be adopted to the budget–limited auction with full knowledge of p as follows (for more details, see [11]). At each time step $1 \leq t \leq T$, Lueker's algorithm chooses a bid $b^+(B_t, t)$ that satisfies

$$b^+(B_t, t) = \max\{b\} \quad \text{s.t.} \quad \sum_{\sigma=0}^{b} p(\sigma)\sigma \leq \frac{B_t}{T-t+1} \tag{6}$$

where B_t is the current residual budget. The efficiency of this algorithm is guaranteed by the following:

Proposition 1 (Theorem 2 from [11]) *Suppose that we have full information about the market price distribution p. Consider the optimal deterministic solution, that has the full information about the sequence of market prices $\{x_t\}$ as well (i.e., it knows the value of each x_t in advance). Given this, the difference between the performance of Lueker's algorithm and that of the optimal deterministic solution is at most $O(\ln T)$.*

The proof can be found in [11]. However, since neither the sequence of $\{x_t\}$ nor p is known in advance, we combine Lueker's algorithm with Zeng's estimator (instead of the KM estimator) in order to learn the market price distribution and determine an efficient bid at the same time. This leads to the LuekerLearn algorithm (see Algorithm 3), that places a bid $b^+(B_t, t)$ as follows:

Algorithm 3 The LuekerLearn Algorithm

1: **Inputs:** $T > 0$, $B > 0$, $\hat{p}_1$ is uniform;
2: **for** $t = 1 \rightarrow T$ **do**
3: place a bid $b^+(B_t, t)$ according to Equation 7;
4: use Zeng's estimator to update $\hat{p}_{t+1}$;
5: **end for**

$$b^+(B_t, t) = \max\{b\} \quad \text{s.t.} \quad \sum_{\sigma}^{b} \hat{p}_t(\sigma)\sigma \leq \frac{B_t}{T-t+1} \tag{7}$$

where $\hat{p}_t$ is the estimate of p at time step $1 \leq t \leq T$, and $b^+(B_T, T) = B_T$. Based on the censored observation o_t, it then updates the estimation of p (i.e., $\hat{p}_{t+1}$), using the Zeng's estimator (Algorithm 3, lines $2-5$). The intuition of the algorithm is that as the estimate $\hat{p}_t$ gets more accurate over time, the algorithm converges to the original algorithm provided by Lueker. Since Proposition 1 guarantees the efficiency of the latter, LuekerLearn can also achieve low regret bounds, as we will prove later in this work.

5 Performance Analysis

Within this section, we analyse the performance of the aforementioned algorithms. In particular, we derive performance regret bounds for each of the algorithms. We also show that these regret bounds imply the fact that the algorithms converge to the theoretical optimal solution with high probability. We start with $\varepsilon-$First:

Theorem 2 *Let $T \geq 8(-\ln\frac{\beta}{2})$ for some $0 < \beta < 1$. For any $0 < \varepsilon < 1$ and $B > \varepsilon CT$, and $T > C$ where C is the support of the market price, the regret of the modified version of $\varepsilon-$*First *where Suzukawa's method is used for the estimation of the market price distribution, is at most $C\varepsilon T + \sqrt{\frac{8(-\ln\frac{\beta}{2})T}{\varepsilon}}$ with probability of at least $(1-\beta)$. In addition, by setting $\varepsilon = \left(\frac{-2\ln\frac{\beta}{2}}{C^2T}\right)^{\frac{1}{3}}$, the regret bound can be refined to $3\left(-2\ln\frac{\beta}{2}\right)^{\frac{1}{3}} C^{\frac{1}{3}}T^{\frac{2}{3}}$.*

Note that the condition $B > \varepsilon CT$ guarantees that within the exploration phase, the bids are uniformly sampled from the entire interval $[1, C]$, since the algorithm samples from the $[1, \frac{B}{\varepsilon T}]$. This condition guarantees that Suzukawa's estimator can fully cover the interval $[1, C]$. In addition, the $O(C^{\frac{1}{3}}T^{\frac{2}{3}})$ regret bound is weak if the $C < T$ condition does not hold. In particular, by fixing T and increasing C, we will get a regret bound that is worse than $O(T)$. Nevertheless, this regret bound achieves Hannan consistency (i.e., sub–linear in T) if $C < T$.

It is also worth to note that since we only consider the case $B \sim O(T)$, $O(T^{\frac{2}{3}})$ regret bound is equiva-

lent to $O(B^{\frac{2}{3}})$, as $T^{\frac{2}{3}} > \frac{B^{\frac{2}{3}}}{C^{\frac{2}{3}}}$. Given the results for $\varepsilon-$First, we now turn to the analysis of GPL and LuekerLearn. If we consider the sequence of the bids as random variables, then the consistency and the zero bias property of KM estimators such as Suzukawa's require independency between the bids and the market price[3]. However, since in both GPL and LuekerLearn we choose the current bid based on the empirical distribution which is built by using the previous observations, it is easy to see that the current bid is not independent from the sequence of the previous market prices. Thus, the consistency (and the convergence rate) of the empirical distribution might not be guaranteed if the standard KM or Suzukawa's estimator is used within GPL and LuekerLearn (for more details, see, e.g., [13, 16]). This implies that neither GPL or LuekerLearn can achieve Hannan–consistency if we use their versions proposed in [1] without any modifications. To overcome this issue, we replace the KM estimator within GPL and LuekerLearn with a novel censored data estimator proposed by Zeng [19]. Due to its complexity and the space limitations, the detailed description of Zeng's estimator is omitted (for more details, see [19]). However, we sketch it as follows.

Zeng's method assumes that there is an underlying set of (known) variables $\mathbf{L}$ that describes the dependency between the two sequences of chosen bids and market prices. Furthermore, suppose that $\mathbf{L}$ is sufficient enough such that for each t, x_t (i.e., the market price) is independent from b_t (i.e., the chosen bid value), conditional to the value of $\mathbf{L}$ at time step t, denoted with $\mathbf{L}_t$. In addition, this method requires that either x_t or b_t follows Cox's proportional model; that is, at least one of the following conditions must hold:

$$p(x_t|\mathbf{L}_t = \mathbf{l}) \sim \lambda_x \exp\{\beta'\mathbf{l}\} \quad (8)$$

$$p(b_t|\mathbf{L}_t = \mathbf{l}) \sim \lambda_b \exp\{\gamma'\mathbf{l}\} \quad (9)$$

for some unknown λ_x, λ_b random variables, and some (unknown) parameters β and γ, respectively. Let $\hat{P}_t$ denote Zeng's estimate of the market price P after t time steps. Zeng proved that $\sqrt{t}(\hat{P}_t - P)$ is a Donsker–class empirical process. Based on this result, we state the following:

Theorem 3 *The abovementioned assumptions hold for both* GPL *and* LuekerLearn*. Given this, by using Zeng's estimation method, the estimate $\hat{P}_t$ of the market price distribution converges in probability to the true distribution P with rate $O(\frac{1}{\sqrt{t}})$ in both* GPL *and* LuekerLearn*.*

This theorem implies the following statements:

[3]In fact, it is sufficient to guarantee that the covariance between the bids and the market price is 0.

Theorem 4 *There exists a constant $K > 0$ that only depends on the market price distribution p, such that the regret of* GPL*, combined with Zeng's estimator, is at most $O(2K\sqrt{T})$ with high probability.*

Similarly, we have the following theorem for LuekerLearn:

Theorem 5 *There exists a constant $K > 0$ that only depends on the market price distribution p, such that the regret of* LuekerLearn*, combined with Zeng's estimator, is at most $O(2K(\sqrt{T} + \ln T))$ with high probability.*

Similarly to the case of $\varepsilon-$First, here we can also transform the regret bounds of GPL and LuekerLearn to $O(2K\sqrt{B})$ and $O(2K(\sqrt{B} + \ln B))$, respectively.

Note that Theorems 4 and 5 imply that GPL converges faster to the optimal solution than LuekerLearn, as T tends to infinity. This is due to the additional $\ln T$ term within the regret bound of LuekerLearn. The reason behind this is that LuekerLearn in fact converges with rate $O(\ln T)$ towards GPL. Hence an additional, $O(\ln T)$, gap is needed here. Also note that by using Zeng's method in $\varepsilon-$First , we would get worse results, compared to Theorem 2, as with the approach from Suzukawa, we could derive exact constant coefficient values for the regret bound, while Zeng's method only provides asymptotic regret bounds. In addition, since in both Theorems 4 and 5, the value of K is typically hard to be identified, the results of these theorems are in fact focussing on the asymptotic behaviour of the algorithms (i.e., both algorithms are Hannan consistant), and do not address whether the bounds are tight.

6 Numerical Evaluation

While we have so far developed theoretical upper bounds for the performance regret of the algorithms, we now turn to practical aspects and examine their performance in a realistic setting,as it might be the case that regret bound for $\varepsilon-$First is not tight, and thus, it might perform better than $O(T^{\frac{2}{3}})$ in many cases, as we will demonstrate later within this section. Given this, in this section, we aim to investigate whether the algorithms achieve high performance when applied to practical sponsored search auction problems. To do so, we first describe our parameter settings in Section 6.1. We then continue with the numerical results of the algorithms' performance in Section 6.2.

6.1 Parameter Settings

To investigate the performance of the algorithms, we use the same dataset as [1], taken from a real-world

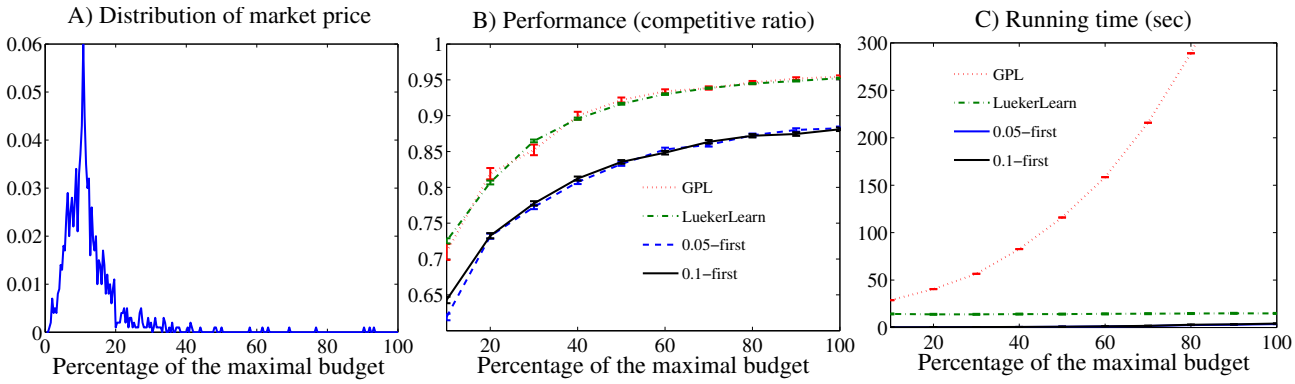

Figure 1: Numerical results on a subset of keywords with single distribution peaks, and with budgets ranging from 10% to 100% of the maximal budget $B_k(T)$: A) A typical single-peaked distribution of the market price. B) The performance of the algorithms, measured in competitive ratio against the optimal solution. C) Computational cost of the algorithms.

sponsored search auction database. Given this, we follow the parameter settings described there. In particular, for each experiment, we use $U = 10$ periods, each of which comprises $T = 100$ auctions and the budget is refilled at the beginning of each period. For a fair comparison to the results of Amin *et al.*, the maximal budget $B_k(T)$ for keyword k is also selected in the same way they do. In particular, we set $B_k(T)$ such that $G^*(B_k(T), T) = fT$ for $f = 0.1$ (i.e., 10%) and $T = 100$. This setting aims to satisfy that, on average, we can win 10% of the auctions. Within each experiment, we vary the budget from $\frac{B_k(T)}{10}$ up to $B_k(T)$ with a step of $\frac{B_k(T)}{10}$. Each experiment was repeated 100 times (for more details of the parameter settings, see [1]). Within our experiments, we run $\varepsilon-$First with $\varepsilon = 0.05$ and $\varepsilon = 0.1$, respectively, as these values are typically more efficient than other value settings[4].

6.2 Numerical Results

Given the description of the parameter settings above, we now investigate the numerical results in more detail. In particular, we observed that the real distribution of the market price can typically be distinguished into two groups. In the first group, the market price usually concentrates at low values, creating a single-peaked distribution (see Figure 1A). Within the second group, the market price is typically more scattered, causing multiple peaks within the distribution (see Figure 2A). The performance efficiency of the algorithms also vary between these distribution groups. Therefore, we distinguish these two cases, and separately examine the performance of the algorithms within these cases. In particular, Figure 1 depicts the numerical results for the single-peaked case, and Figure 2 depicts the results for the multi-peaked case, respectively (here, the second group typically contains two peaks, as is also shown in Figure 2).

We first evaluate the single-peaked case (Figure 1). As mentioned earlier, Figure 1A shows the distribution of the market price. In addition, Figure 1B plots the performance of the algorithm, compared against that of the optimal stochastic solution described in Section 3. Here, the optimal stochastic solution also uses an MDP model to determine the optimal bidding policy, but assuming full knowledge of the distribution of market prices. Figure 1C depicts the running time of each algorithm. As can be seen from the figures, GPL and LuekerLearn provide similar performance, and both outperform the two versions of $\varepsilon-$First, 0.05–First and 0.1–First, by up to 10%. The reason for this is that since the market price is typically concentrated at low values, all the algorithms can quickly learn this. This allows GPL and LuekerLearn to use small bids to refine the estimation of the market price distribution at small values, and thus, to bid more efficiently. In contrast, as $\varepsilon-$First stops learning after the exploration phase, its estimation at the small values is not as accurate as the others'. Given this, $\varepsilon-$First bids suboptimally in more time steps, compared to the other two. Nevertheless, note that $\varepsilon-$First can still achieve by up to 88% of the optimal solution.

On the other hand, the running time of GPL is significantly larger, compared to that of the others (Figure 1C). In particular, GPL typically needs more than 500 seconds to evaluate the case of maximal budget $B_k(T)$, while $\varepsilon-$First algorithms only need less than 10 seconds. The reason for this is that GPL recomputes the MDP for the optimal decision at each step, after updating its price distribution. This is

[4]Note that all the numerical tests appearing in this paper are performed on a personal computer, Intel® Xeon® CPU W3520 @2.67GHz with 12GB RAM and under Windows 7 operating system. The code was written and tested on Matlab R2012a.

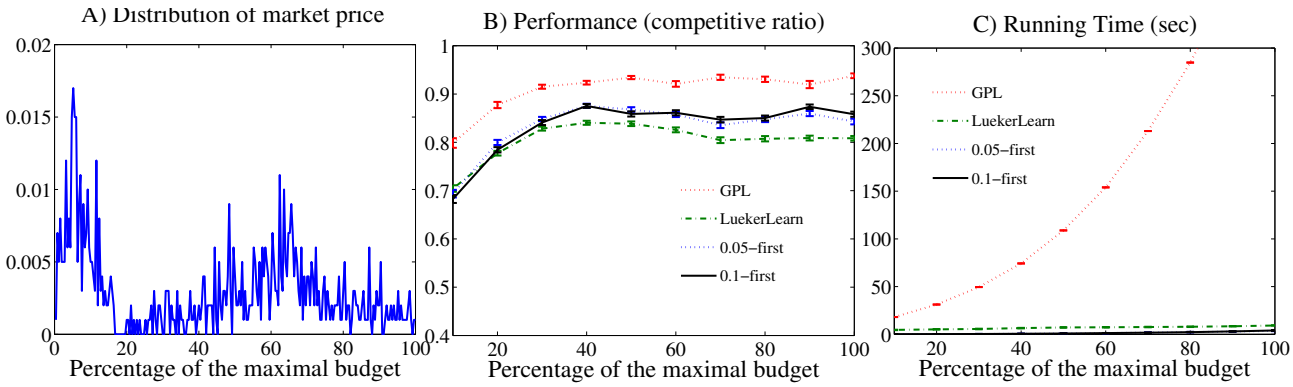

Figure 2: Numerical results on a subset of keywords with more than one distribution peaks, and with budgets ranging from 10% to 100% of the maximal budget $B_k(T)$: A) A typical two-peaked distribution of the market price. B) The performance of the algorithms, measured in competitive ratio against the optimal solution. C) Computational cost of the algorithms.

computationally expensive, especially for large budgets. By contrast, the $\varepsilon-$First algorithms only compute the MDP once, at the end of the exploration phase. Thus, despite its best competitive ratio performance, the running time of GPL would limit its suitability for real-time deployment. LuekerLearn also needs approximately 15 seconds to solve this problem instance. Given this, for single-peaked distributions, LuekerLearn yields the best trade-off between efficiency and computational cost, as it achieves similar performance to that of the GPL (33 times faster), and is almost as fast as the $\varepsilon-$First algorithms.

Within the case of distributions with multiple peaks (in this case, we consider the two-peaked version), we can see that GPL still provides the best performance (see Figure 2B). However, in this setting, $\varepsilon-$First outperforms LuekerLearn by approximately 5%. The reason behind this is that due to multiple peaks, LuekerLearn starts to deviate between the peaks, as it makes more observations (see [1] for more details). This implies that LuekerLearn makes more suboptimal bids, as placing bids at the first peak is typically more desirable, as opposed to the bids close to the second peak. On the other hand, due to its restricted learning phase, $\varepsilon-$First typically learns the values around the first peak, and thus, can act more efficiently, compared to LuekerLearn. Nevertheless, both $\varepsilon-$First and LuekerLearn still achieve good performance, as both typically provide at least 80% of the optimal solution's.

In terms of computational cost, GPL still requires the highest running time (more than 600 seconds for the case of maximal budget $B_k(T)$). By contrast, both $\varepsilon-$First and LuekerLearn require at most 10 seconds. Note that $\varepsilon-$First is typically two times faster than LuekerLearn. Therefore, in the two-peaked case, $\varepsilon-$First is clearly the best choice for the budget-limited auction problem, as it provides good performance (above 85% of the optimal solution), and achieves by far the lowest computational cost.

7 Conclusions

We studied the online bid optimisation problem in budget-limited sponsored search auctions, where the market price is drawn from a fixed, but unknown distribution, and is censored by the value of our current bid. Although existing algorithms have been shown to achieve good performance in practice, no theoretical performance analysis has been provided for this problem. Given this, we proposed $\varepsilon-$First, and we show that it provably achieves $O(T^{\frac{2}{3}})$ regret bound with high probability, where T is the number of total auctions. We also provided an affirmative answer to the research question raised in [1], which conjectures that GPL, a state-of-the-art algorithm for the budget-limited sponsored search auction problem, can achieve asymptotically optimal performance. In particular, we proved that GPL achieves $O(\sqrt{T})$ regret bound with high probability. We also showed in the paper that the regret bound of LuekerLearn, another state-of-the-art algorithm, is $O(\sqrt{T} + \ln T)$, also with high probability. In addition, we compared the performance of the algorithms on real-world data, and observed that, although GPL provides the highest performance, it is by far the most computationally expensive algorithm, and its running time would make it infeasible for real time deployment. On the other hand, LuekerLearn would be the best choice in the case of single-peaked distributions, as it provides the best trade-off between efficiency and computational cost. For the two-peaked distribution case, we showed that $\varepsilon-$First outperforms LuekerLearn with a reduced running time.

References

[1] Amin, K., Kearns, M., Key, P., and Schwaighofer, A. (2012). Budget optimization for sponsored search: Censored learning in MDPs. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, pages 54–63.

[2] Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2013). Bandits with knapsacks. In *IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216.

[3] Berg, J., Greenwald, A., Naroditskiy, V., and Sodomka, E. (2010). A first approach to autonomous bidding in ad auctions. In *Workshop on Trading Agent Design and Analysis at the 11th ACM Conference on Electronics Commerce*.

[4] Engel, Y. and Chickering, D. M. (2008). Incorporating user utility into sponsored-search auctions. *Proceedings of the Seventh International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1565–1569.

[5] Even-Dar, E., Mannor, S., and Mansour, Y. (2002). PAC bounds for multi-armed bandit and Markov decision processes. In *COLT*.

[6] Feldman, J., Muthukrishnan, S., Pal, M., and Stein, C. (2007). Budget optimization in search-based advertising auctions. In *Proceedings of the 8th ACM conference on Electronic commerce*, EC '07, pages 40–49. ACM.

[7] Gummadi, R., Key, P., and Proutiere, A. (2012). Optimal bidding strategies and equilibria in dynamic auctions with budget constraints. *Available at SSRN: http://ssrn.com/abstract=2066175*.

[8] Jordan, P. R., Wellman, M. P., and Balakrishnan, G. (2010). Strategy and mechanism lessons from the first ad auctions trading agent competition. In *Proceedings of the 11th ACM conference on Electronic commerce*, EC '10, pages 287–296, New York, NY, USA. ACM.

[9] Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Society*, **53**, 457–481.

[10] Kitts, B. and Leblanc, B. (2004). Optimal bidding on keyword auctions. *Electronic Markets*, **14**(3), 186–201.

[11] Lueker, G. S. (1995). Average-case analysis of off-line and on-line knapsack problems. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, SODA '95, pages 179–188, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

[12] Pardoe, D. and Stone, P. (2011). A particle filter for bid estimation in ad auctions with periodic ranking observations. *Proceedings of the Tenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 687–694.

[13] Phadia, E. and Van Ryzin, J. (1980). A note on convergence rates for the product limit estimator. *The Annals of Statistics*, **8**(3), 673–678.

[14] Stavrogiannis, L. C., Gerding, E. H., and Polukarov, M. (2013). Competing intermediary auctions. *Proceedings of the Twelfth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 667–674.

[15] Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

[16] Suzukawa, A. (2004). Unbiased estimation of functionals under random censorship. *Journal of the Japan Statistical Society*, **32**(2), 153–172.

[17] Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers.

[18] Tran-Thanh, L., Chapman, A., Rogers, A., and Jennings, N. R. (2012). Knapsack based optimal policies for budget-limited multi-armed bandits. *Proceedings of the 26th Conference on Artificial Intelligence (AAAI 2012)*, pages 1134–1140.

[19] Zeng, D. (2004). Estimating marginal survival function by adjusting for dependent censoring using many covariates. In *The Annals of Statistics*, pages 1533–1555.

[20] Zhou, Y. and Naroditskiy, V. (2008). Algorithm for stochastic multiple-choice knapsack problem and keywords bidding. In *WWW08: Workshop on Targeting and Ranking for Online Advertising*.