Optimizing reserve prices for publishers in online ad auctions

Optimizing reserve prices for publishers in online ad auctions

Jason Rhuggenaath, Alp Akcay, Yingqian Zhang and Uzay Kaymak
School of Industrial Engineering
Eindhoven University of Technology
Eindhoven, The Netherlands
Email: {j.s.rhuggenaath, a.e.akcay, yqzhang, u.kaymak}@tue.nl

***Abstract*—In this paper we consider an online publisher that sells advertisement space and propose a method for learning optimal reserve prices in second-price auctions. We study a limited information setting where the values of the bids are not revealed and no historical information about the values of the bids is available. Our proposed method is based on the principle of Thompson sampling combined with a particle filter to approximate and sample from the posterior distribution. Our method is suitable for non-stationary environments, and we show that, when the distribution of the winning bid suffers from estimation uncertainty, taking the gap between the winning bid and second highest bid into account leads to better decisions for the reserve prices. Experiments using real-life ad auction data show that the proposed method outperforms popular bandit algorithms.**

***Index Terms*—Machine learning; Stochastic optimization; Auctions; Thompson sampling; Multi-armed bandits.**

I. Introduction

One of the main mechanisms that web publishers use in online advertising in order to sell their advertisement space is the real-time bidding (RTB) mechanism [1]. In RTB there are three main platforms: supply side platforms (SSPs), demand side platforms (DSPs) and an ad exchange (ADX) which connects SSPs and DSPs. The SSPs collect inventory of different publishers and thus serve the supply side of the market. Advertisers which are interested in showing online advertisements are connected to DSPs. When a user visits a webpage with an advertisement (ad) slot, the publisher sends a request to the ADX (via an SSP) indicating that an impression can potentially be displayed in this particular ad slot. At the same time, advertisers that are connected to DSPs send bid requests to the ADX indicating that they are willing to bid for this impression. A real-time auction then decides which advertiser is allowed to display its ad and the amount that the advertiser needs to pay. The most popular auction mechanism is the second-price auction, where the winning advertiser pays the second highest bid in the auction.

Publishers can set a reserve price for their inventory in the second-price auction. Due to the reserve price, all bids below the reserve price are disregarded, and as a consequence, there is a possibility that the ad slot is not sold. If the auction does have a winner, the winner pays the maximum of the second highest bid and the reserve price. In this paper we take the perspective of an online publisher that submits his inventory of advertisement space to an SSP and needs to decide on the optimal value of the reserve price. We assume that the publisher has limited information about the winning bid and second highest bid in the auction. More specifically, the publisher does not observe the actual values of the winning bid and second highest bid. After each sale attempt on the RTB market, the publisher only knows whether the sale was successful and the revenue that is received from that sale. This setting is relevant for publishers that are small and medium size enterprises (SMEs), since the ADX and the connected SSPs typically do not reveal the actual bids placed in the auction but only the result of the auction. Due to the limited feedback, the publisher faces an exploration-exploitation trade-off. He needs to experiment with different reserve prices to figure out which one works best, but at the same time, he does not want to explore too much since he wants to use the best reserve price as much as possible (exploitation). In this paper we present a method that addresses the problem of the publisher. Our method is based on the principle of Thompson sampling combined with a particle filter to approximate and sample from the posterior distribution. In addition we show that modeling both the winning bid and second highest bid leads to superior performance compared to just tracking the winning bid.

We summarize the main contributions of this paper as follows:

- We propose a method for learning optimal reserve prices in a limited information setting where the values of the bids are not revealed and no historical information about the values of the bids is available.
- Our proposed method is suitable for non-stationary environments.
- We show that, when the distribution of the winning bid suffers from estimation uncertainty, taking the gap between the winning bid and second highest bid into account leads to better decisions for the reserve prices.
- Experiments using real-life ad auction data show that the proposed method outperforms popular bandit algorithms.

The remainder of this paper is organized as follows. In Section II we discuss the related literature. Section III provides

a formal formulation of the problem. In Section IV we present the our proposed method for setting reserve prices that only uses a model for the winning bid. In Section V, we extend the initial model by modeling the second highest bid. In Section VI we perform experiments and compare our method with baseline strategies in order to assess the quality of our proposed method. Section VII concludes our work and provides some interesting directions for further research.

II. Related literature

The problem of maximizing revenues in online advertising has received increasing attention in the machine learning literature over the last decade (see e.g. [1], [2], [3], [4], [5]). Some works such as [6], [7], [8] try to estimate or predict the winning bid (possibly in the presence of censoring) by using a historical dataset containing the top two bids and additional features relating to the context of the user visiting a website. These papers use different methods in order to predict the winning bid. In [6] a deep neural network is used, in [7] a regression model with a gamma distribution is used to deal with censoring, and in [8] a mixture model is used. These studies however do not focus on setting optimal reserve prices. In [9] an online learning approach is used to derive a policy for setting optimal reserve prices, but the analysis makes the assumption that the environment is stationary. Other works such as [2], [10], [11], [12] use historical data to directly predict the optimal reserve price or the winning bid (which can indirectly be used to set a reserve price). A drawback of using models based on historical data is that they may not perform well in non-stationary environments. Most of the studies mentioned above do not set reserve prices in an adaptive way that adjusts to changing environments. Some studies such as [2], [3], [4] do study adaptive reserve prices, but they assume that the winning bid and/ or second highest bid are observed. In this paper we do not make this assumption. Other works such as [13], [14], [15] make use of strategies based on Multi-armed Bandit (MAB) models in order to optimize revenues in RTB. Most of these studies focus on the problem of click-through-rate (CTR) prediction and ad placement optimization. A related work in terms of methodology is [16], where the particle filter is also used in combination with Thompson sampling. However, [16] studies a different problem, namely optimizing revenues for an SSP using a header-bidding strategy. To summarize, the main differences between this paper and previous works are that: (i) we show how to set adaptive reserve prices in possibly non-stationary environments; (ii) we do not assume that the publisher observes the top two bids in the ad auction, but only observes the revenue of each auction.

III. Problem Statement

We consider a publisher that owns a single advertisement slot and that there is a sequence of impressions (corresponding to this advertisement slot) arriving over time. Time is discretized, and time periods are denoted by $t \in \mathbb{N}$. At the beginning of each time period (that is, upon arrival of an impression) the publisher has to decide on a reserve price $p_t \in [p_l, p_h]$. The prices $0 < p_l < p_h$ are the minimum and maximum reserve prices that are acceptable to the publisher. After deciding on a reserve price the impression is offered for sale on the RTB-market via a Supply Side Platform (SSP). The SSP runs a second-price auction for the impression and the revenue of the publisher depends on the outcome of this auction. Let X_t and Y_t denote the highest and second highest bid respectively in the auction for impression at time t. Then the revenue (or return) of the publisher at time t is given by $R_t = \mathbb{I}\{p_t \leq X_t\} \cdot \max\{Y_t, p_t\}$. Here $\mathbb{I}\{A\} = 1$ if A is true and $\mathbb{I}\{A\} = 0$ otherwise. The expression for R_t says that if the reserve price p_t is too high ($p_t > X_t$) then the publisher receives zero revenue. If the reserve price is not too high ($p_t \leq X_t$) then the revenue equals the maximum of the second highest bid and the reserve price. Note that, in general, the publisher does not observe the value of X_t and Y_t after a (successful) sale.

Assumption 1: If a sale was not successful, that is, if $p_t > X_t$, then the publisher does not observe X_t and Y_t. If a sale was successful, that is, if $p_t \leq X_t$, then the publisher observes $\max\{Y_t, p_t\}$.

The objective of the publisher is to maximize the cumulative revenue over the sales horizon of length T. Thus the revenue optimization problem over T time periods or impressions can be expressed as follows:

$$\max_{p_1,\ldots,p_T} \mathbb{E}\left\{\sum_{t=1}^{T} \mathbb{I}\{p_t \leq X_t\} \cdot \max\{Y_t, p_t\}\right\} \tag{1}$$

Remark 1: In the literature on online advertising and the RTB-market, the reserve price is sometimes also referred to as the *floor price*. In the remainder of this paper we will use the term *top bid* to refer to the winning bid in the online auction and we will use the term *second bid* to refer to the second highest bid.

Remark 2: In order to simplify the exposition of our method, we focus on the case where there is a single ad slot. However, in practice, the publisher may want to set a reserve price depending on the characteristics of the user and the ad slot. Our method can also be applied in such a setting by, for example, making segments of users and applying our method for each segment.

Remark 3: We will use the symbol $\wedge$ to denote the operator for the logical conjunction and the symbol $\vee$ to denote the operator for the logical disjunction.

IV. Initial Model: TS-PF-TB

In this section we present a method to sequentially optimize the reserve price p_t. It combines the principle of Thompson sampling [17] with an approximation based on a particle filter. We refer to the model of this section as TS-PF-TB (Thompson sampling with particle filter and only modeling the top bid).

A. Parametric model for Top Bid

Let f_θ^X denote a family of distributions for the top bid X parametrized by θ and let F_θ^X denote the corresponding cumulative distribution function (CDF). We assume that the

top bid at time t, X_t, is a draw from the distribution f_θ^X. The underlying parameter vector is modeled as a random variable with a prior distribution $\pi_0^X(\theta)$. The posterior distribution at the end of time period t is given by:

$$\pi_t^X(\theta) \propto \prod_{i=1}^{t} \ell_i \times \pi_0^X(\theta) \tag{2}$$

Here $\ell_i = F_\theta^X(p_i) \cdot \mathbb{I}\{p_i > X_i\} + \big(1 - F_\theta^X(p_i)\big) \cdot \mathbb{I}\{p_i \leq X_i\}$ is the likelihood of the observed result of the auction at time i if the top bid X_i is a draw from the distribution f_θ^X.

Our method follows the following main steps:

- Sample a value θ from the posterior distribution at time t, $\pi_t^X(\theta)$. This is the Thompson sampling step [17].
- Maximize the objective function of the publisher assuming that $X_{t+1} \sim f_\theta^X$.
- Observe the outcome of the auction for time $t+1$, that is, observe $\mathbb{I}\{p_{t+1} \leq X_{t+1}\}$ and update the posterior distribution to get $\pi_{t+1}^X(\theta)$.

B. Objective function

Note that the expectation in Equation (1) depends on the joint distribution of the top two bids (X_t and Y_t). As the top two bids are not observed it is difficult to model and optimize the objective in Equation (1) directly. Therefore we use the following objective instead, which only depends on the top bid (X_t):

$$\max_{p_1,\ldots,p_T} \mathbb{E}\left\{\sum_{t=1}^{T} \mathbb{I}\{p_t \leq X_t\} \cdot p_t\right\} \tag{3}$$

The objective value of the maximization problem in (3) is a lowerbound for the objective value of (1), we use it because it is easier to evaluate. Furthermore, by using (3), we can use the update rule given by (2). Note that the maximization problem in (3) decomposes into smaller problems that can be solved in each period. More specifically, the optimization problem that needs to be solved in period t is

$$\max_{p_t} \; (1 - F_\theta^X(p_t)) \cdot p_t \tag{4}$$

C. Sampling and updating of the posterior distribution: particle filter method

In order to sample from and update the posterior distribution we use the particle filter method [18]. The main idea in the particle filter method is to approximate the sequence of posterior distributions that is generated over time by a sequence of discrete distributions on K points (called particles). The posterior distribution using information until time t given by $\pi_t(\theta)$ is approximated by a discrete distribution on K points $\theta_{t,1}, \ldots, \theta_{t,K}$ with weights given by $w_{t,1}, \ldots, w_{t,K}$. The sequence of discrete distributions are derived from each other through a evolution step. Furthermore, in order to manage the degeneracy of the weights in the approximation, a selection step is also included. We will elaborate on the evolution step and selection step below.

The main motivation for using the particle filter is that it allows us to deal with non-stationarity of the parameter vector θ. Until now we have assumed that $X_t \sim f_\theta = f_{\theta_t} \forall t$. However, it might be the case that θ changes over time. The particle filter allows us to model this change directly. More specifically, we assume that $\theta_{t+1} = \theta_t + \epsilon_t$, where ϵ_t is a drift term with mean zero. This specification models a situation where the parameter vector takes a step in an unknown direction and captures the possibility that θ changes over time.

a) Evolution step (updating the posterior distribution): This step consists of updating the weights and particles that are to be used in the next time period. In the evolution step the particles at time $t+1$ are derived from the particles at time t and the weights at time $t+1$ are derived from the weights at time t. We assume that the true underlying parameter vector θ_t evolves according to Markovian dynamics described by a transition kernel $P(\theta_{t+1}|\ \theta_t)$. The new particles at time $t+1$ are sampled from a proposal distribution $Q(\theta_{t+1,k}|\ \theta_{t,k})$. Suppose that we are approximating the posterior distribution of the parameter vector for the top bid. Then given these transition kernels, the unnormalized weights using information until time $t+1$, are updated based on the result of the last auction (the likelihood of the result) at time $t+1$:

$$w_{t+1,k} = \ell_{t+1,k} \times \frac{P(\theta_{t+1}|\ \theta_t)}{Q(\theta_{t+1,k}|\ \theta_{t,k})} \times w_{t,k} \tag{5}$$

where,

$$\begin{aligned} \ell_{t+1,k} =& F^X_{\theta_{t+1,k}}(p_{t+1}) \cdot \mathbb{I}\{p_{t+1} > X_{t+1}\} \\ &+ \Big(1 - F^X_{\theta_{t+1,k}}(p_{t+1})\Big) \cdot \mathbb{I}\{p_{t+1} \leq X_{t+1}\}. \end{aligned} \tag{6}$$

Here $F^X_{\theta_{t,k}}$ is the CDF of the top bid using information until time t with parameter vector $\theta_{t,k}$. Essentially, the weight of particle k is adjusted proportionally based on the likelihood of observing the result of the auction at time $t+1$ under the assumption that the CDF of the top bid has parameter vector $\theta_{t+1,k}$. In our implementation we take $P(\theta_{t+1}|\ \theta_t) = Q(\theta_{t+1,k}|\ \theta_{t,k})$ so that the update rule simplifies to

$$w_{t+1,k} = \ell_{t+1,k} \times w_{t,k}. \tag{7}$$

After updating the weights, they are normalized so that they sum up to 1.

b) Selection step (dealing with weight degeneracy): This step is needed in order to deal with the possibility of weight degeneracy. From the update rule in Equation (7) it follows that the new weights can only decrease after each update and at some point many of the weights in the approximation can become close to zero (degeneracy of weights). In our implementation we deal with this problem in two ways. In the first method we adapt the update rule in Equation (7) by introducing an offset constant $0 < c < 1$ as follows

$$w_{t+1,k} = (c + \ell_{t+1,k}) \times w_{t,k}. \tag{8}$$

With this specification the weights can both increase and decrease depending on the value of $\ell_{t+1,k}$. This reduces the possibility of weight degeneracy. In our implementation we set $c = 0.5$.

In the second method we use the strategy defined in [19]. We compute $S = (\sum_{k=1}^{K} w_{t,k}^{l})^{-1}$ to measure the degree of degeneracy of the particle filter. If $S < S_{\min}$ then we resample all the particles by sampling K times with replacement from the current set of particles. Afterwards, we reset the weight of particle k to $w_{t+1,k} = 1/K$. Here $S_{\min}$ and l are hyperparameters of the particle filter.

D. Implementation details particle filter

We model the top bid X_t in the ad auction with a normal distribution with parameter vector $\theta = \left(\theta^{(1)}, \theta^{(2)}\right)$, where $\theta^{(1)} = \mu$ and $\theta^{(2)} = \sigma > 0$. Here μ is the mean and σ is the standard deviation of the normal distribution.

The particle filter works as follows:

1) We assume that the parameter vector evolves according to Markovian dynamics such that $\ln\left(\theta_t^{(2)}\right) = \ln\left(\theta_{t-1}^{(2)}\right) + \epsilon^2$ and $\theta_t^{(1)} = \theta_{t-1}^{(1)} + \epsilon^1$. Here $\epsilon^1, \epsilon^2 \sim \mathcal{N}(0, \kappa)$ are independently normally distributed random variables with mean zero and standard deviation κ. This completes the specification of $P(\theta_{t+1} |\ \theta_t)$.
2) At each time period t we use K particles denoted by $\theta_{t,k}$ for $k = 1, \ldots, K$.
3) The particles evolve according to the same Markovian dynamics as the unknown parameter vector such that $\ln\left(\theta_{t,k}^{(2)}\right) = \ln\left(\theta_{t-1,k}^{(2)}\right) + \epsilon^2$ and $\theta_{t,k}^{(1)} = \theta_{t-1,k}^{(1)} + \epsilon^1$. Here $\epsilon^1, \epsilon^2 \sim \mathcal{N}(0, \kappa)$ are independently normally distributed random variables. This completes the specification of $Q(\theta_{t+1,k} |\ \theta_{t,k})$.
4) We use a uniform distribution for the prior of the initial parameter vector θ_t at time $t = 0$. That is, for each $k = 1, \ldots, K$ we generate the components of particle k by drawing from a uniform distribution: $\theta_{0,k}^{(1)} \sim \mathcal{U}\left(\theta_{LB}^{(1)}, \theta_{UB}^{(1)}\right)$ and $\theta_{0,k}^{(2)} \sim \mathcal{U}\left(\theta_{LB}^{(2)}, \theta_{UB}^{(2)}\right)$.
5) In our implementation we set $S_{\min} = K/2.5$ and $l = 5$.

E. Batch updating and Price smoothing

In practice the true realizations of Y_t and X_t can be quite noisy even in the case where there might be an upwards or downwards trend. As the publisher can only learn about the distribution of the bids by updating his prior *after* each sale, the noisy realizations might distract the algorithm from learning the underlying trend. In order to set reserve prices that are robust to noise we employ two methods: *batch updating* and *price smoothing*. In the first method (batch updating), we only update the reserve price every M time periods using the information collected in time periods $t-M+1$ to t. In this way, the effectiveness of the reserve price that is asked between periods $t - M + 1$ to t can be estimated more accurately compared to the situation where the reserve price changes every period. When performing the batch update, we use the following variant of Equation (8):

$$w_{t+M,k} = \prod_{i=t+1}^{t+M} (c + \ell_{i,k}) \times w_{t,k}. \tag{9}$$

The particles evolve according to $Q(\theta_{t+M,k} |\ \theta_{t,k})$:

$$\theta_{t+M,k}^{(1)} = \theta_{t,k}^{(1)} + \epsilon^1,\ \epsilon^1 \sim \mathcal{N}(0, \kappa) \tag{10}$$

$$\ln\left(\theta_{t+M,k}^{(2)}\right) = \ln\left(\theta_{t,k}^{(2)}\right) + \epsilon^2,\ \epsilon^2 \sim \mathcal{N}(0, \kappa) \tag{11}$$

In the second method (price smoothing), we average the reserve price that is recommended in period t with the reserve prices asked in the previous L time periods. More specifically, if the algorithm indicates that p_t^* should be used in period t, we instead average p_t^* with the previous L reserve prices that were used and we instead use $\bar{p}_t = \left(p_t^* + \sum_{k=t-L}^{t-1} p_t\right)/L+1$. The full procedure for learning the reserve prices is described in Algorithm 1.

Algorithm 1 Pseudocode for TS-PF-TB

Require: K, M, L, $\theta_{X,LB}^{(1)}$, $\theta_{X,UB}^{(1)}$, $\theta_{X,LB}^{(2)}$, $\theta_{X,UB}^{(2)}$, p_l, p_h.

Initialize particle filter.

1: Set $t = 0$.
2: Draw initial particles for top bid $\left\{\left(\theta_{X,t,k}^{(1)}, \theta_{X,t,k}^{(2)}\right)\right\}_{k=1}^{K}$ according to: $\theta_{X,t,k}^{(1)} \sim \mathcal{U}\left(\theta_{X,LB}^{(1)}, \theta_{X,UB}^{(1)}\right)$ and $\theta_{X,t,k}^{(2)} \sim \mathcal{U}\left(\theta_{X,LB}^{(2)}, \theta_{X,UB}^{(2)}\right)$.
3: Set initial weights of particles for top bid: $w_{t,k}^{X} = 1/K$, for $k = 1, \ldots, K$.
4: Set $t = t + 1$.

Sample from posterior distribution.

5: Sample θ^X from the posterior distribution after $t-1$ periods, $\pi_{t-1}^{X}(\theta)$, using $\left\{\left(\theta_{X,t-1,k}^{(1)}, \theta_{X,t-1,k}^{(2)}, w_{t-1,k}^{X}\right)\right\}_{k=1}^{K}$.

Price smoothing.

6: Set $\bar{p}_t = \left(p_t^* + \sum_{k=t-L}^{t-1} p_k\right)/L + 1$.

Apply reserve price.

7: Use $\bar{p}_t$ as reserve price in periods $t, t+1, \ldots, t+M-1$.

Update posterior distribution (batch update).

8: Observe $\{\mathbb{I}\{\bar{p}_k \le X_k\}, \max\{Y_k, \bar{p}_k\}\}_{k=t}^{t+M-1}$.
9: Update $\pi_{t-1}^{X}(\theta)$ using Equations (9) - (11) to get $\pi_{t+M-1}^{X}(\theta)$.
10: Set $t = t + M$ and go to Line 5.

V. Extended Model: TS-PF-RAP

The previous section discussed the initial model (TS-PF-TB) based on a model for the top bid. This section improves the performance of the initial model by exploiting the setting of the second-price auction.

A. Modeling the second bid

Note that, based on the feedback that the publisher receives after each sale, he can also maintain and update a distribution of the second highest bid Y_t. Why would the publisher be interested in modeling Y_t in addition to X_t? The main reason for modeling Y_t is related to estimation uncertainty. If the distribution of X_t was known in advance, then the optimal decision based on (4) is easy to obtain and will perform well. However, when there is uncertainty regarding the distribution of the top bid, a small error in the estimation can lead to a revenue of zero if the reserve price is set too high.

By modeling the second bid as well, the publisher can make an inference as to whether the gap between Y_t and X_t is likely to be large or not. If the gap is small, then the publisher could

choose to set his reserve price based on the distribution of Y_t instead of X_t in order to reduce the risk of setting the reserve price too high. If the gap is relatively large, then the publisher can use his estimate of the second bid as a reference point and set a reserve price somewhere between his estimate of Y_t and of X_t, depending on how much risk he is willing to take. Letting f_θ^Y denote a family of distributions for the second bid Y parametrized by θ and letting F_θ^Y denote the corresponding CDF, we can follow the same steps as for the distribution of the top bid. We can specify a prior for the parameter vector of the distribution of the second bid, we can update this prior with the result of the auctions in order to calculate the posterior. In order to sample from the posterior and handle possible non-stationarity of the parameters, we can again use the particle filter as described previously in Section IV-C, IV-D and IV-E. The updating of the weights in the particle filter for the second bid is slightly different now, as it depends on the result of the auction. Note that after observing the result of the auction at time t there are three types of feedback possible. If the floorprice p_t is too high ($p_t > X_t$) then the publisher can deduce that $p_t > Y_t$ holds. If the floorprice is not too high ($p_t \leq X_t$) then the revenue equals the maximum of the second highest bid and the floor price ($R_t = \max\{Y_t, p_t\}$). If $R_t = p_t$, then the publisher can deduce that $p_t \geq Y_t$ and otherwise the publisher can deduce that $p_t \leq Y_t$. In order to update the weights of the particles we use a variant of Equation (8) but with $\ell_{t+1,k}$ replaced by

$$\begin{aligned}\ell_{t+1,k}^Y =& F_{\theta_{t+1,k}^Y}^Y(p_{t+1}) \cdot \mathbb{I}\{A \vee B\} \\ &+ \left(1 - F_{\theta_{t+1,k}^Y}^Y(p_{t+1})\right) \cdot \mathbb{I}\{C\}.\end{aligned} \tag{12}$$

Here $A = (p_{t+1} > X_{t+1})$, $B = (p_{t+1} \leq X_{t+1}) \wedge (R_{t+1} = p_{t+1})$ and $C = (p_{t+1} \leq X_{t+1}) \wedge (R_{t+1} \neq p_{t+1})$. For the batch update, we use the following variant of Equation (9):

$$w_{t+M,k} = \prod_{i=t+1}^{t+M} \left(c + \ell_{t+1,k}^Y\right) \times w_{t,k}. \tag{13}$$

B. Risk-aware pricing

After updating his prior for the distribution of Y_t and X_t the publisher needs to decide on a reserve price for period t. There are in general many ways to accomplish this task. In this paper we present a simple scheme that performs well in our numerical experiments. The intuition behind the scheme is as follows: (i) if the gap between Y_t and X_t is believed to be large, then try to set a reserve price above the second bid (but not too high); (ii) if the gap between Y_t and X_t is believed to be small, then choose a reserve price close to Y_t.

In order to quantify the gap between Y_t and X_t we look at the reserve prices that would maximize the per period revenue in Equation (4). More specifically, we determine the reserve price p_t^X that optimizes Equation (4) using the posterior of X_t and the reserve price p_t^Y that optimizes Equation (4) using the posterior of Y_t. If $|p_t^X - p_t^Y|/p_t^Y \leq \alpha$ then we consider the gap to be small and the publisher sets the reserve price according to $p_t^* = p_t^Y$. If on the other hand $|p_t^X - p_t^Y|/p_t^Y > \alpha$, then the gap is considered to be large enough and the publisher sets a reserve price according to $p_t^* = \omega p_t^X + (1-\omega)p_t^Y$ for some $\omega \in (0,1)$. By controlling the parameters α and ω the publisher can decide the degree to which he wants to exploit the fact that the gap between Y_t and X_t is large.

We refer to the method of this section as TS-PF-RAP (Thompson sampling with particle filter and risk-aware pricing). The full procedure for learning the reserve prices is described in Algorithm 2.

Algorithm 2 Pseudocode for TS-PF-RAP

Require: K, M, L, α, ω, $\theta_{X,LB}^{(1)}$, $\theta_{X,UB}^{(1)}$, $\theta_{X,LB}^{(2)}$, $\theta_{X,UB}^{(2)}$, $\theta_{Y,LB}^{(1)}$, $\theta_{Y,UB}^{(1)}$, $\theta_{Y,LB}^{(2)}$, $\theta_{Y,UB}^{(2)}$, p_l, p_h.

Initialize particle filter.

1: Set $t = 0$.

2: Draw initial particles for top bid $\left\{\left(\theta_{X,t,k}^{(1)}, \theta_{X,t,k}^{(2)}\right)\right\}_{k=1}^K$ according to: $\theta_{X,t,k}^{(1)} \sim \mathcal{U}\left(\theta_{X,LB}^{(1)}, \theta_{X,UB}^{(1)}\right)$ and $\theta_{X,t,k}^{(2)} \sim \mathcal{U}\left(\theta_{X,LB}^{(2)}, \theta_{X,UB}^{(2)}\right)$.

3: Draw initial particles for second bid $\left\{\left(\theta_{Y,t,k}^{(1)}, \theta_{Y,t,k}^{(2)}\right)\right\}_{k=1}^K$ according to: $\theta_{Y,t,k}^{(1)} \sim \mathcal{U}\left(\theta_{Y,LB}^{(1)}, \theta_{Y,UB}^{(1)}\right)$ and $\theta_{Y,t,k}^{(2)} \sim \mathcal{U}\left(\theta_{Y,LB}^{(2)}, \theta_{Y,UB}^{(2)}\right)$.

4: Set initial weights of particles for top bid: $w_{t,k}^X = 1/K$, for $k = 1, \ldots, K$.

5: Set initial weights of particles for second bid: $w_{t,k}^Y = 1/K$, for $k = 1, \ldots, K$.

6: Set $t = t + 1$.

Sample from posterior distribution.

7: Sample θ^X from the posterior distribution after $t-1$ periods, $\pi_{t-1}^X(\theta)$, using $\left\{\left(\theta_{X,t-1,k}^{(1)}, \theta_{X,t-1,k}^{(2)}, w_{t-1,k}^X\right)\right\}_{k=1}^K$.

8: Sample θ^Y from the posterior distribution after $t-1$ periods, $\pi_{t-1}^Y(\theta)$, using $\left\{\left(\theta_{Y,t-1,k}^{(1)}, \theta_{Y,t-1,k}^{(2)}, w_{t-1,k}^Y\right)\right\}_{k=1}^K$.

Risk-aware pricing.

9: Determine p_t^X that optimizes Equation (4) assuming that $X_t \sim f_{\theta^X}^X$.

10: Determine p_t^Y that optimizes Equation (4) assuming that $Y_t \sim f_{\theta^Y}^Y$.

11: **if** $|p_t^X - p_t^Y|/p_t^Y \leq \alpha$ **then**

12: set $p_t^* = p_t^Y$

13: **else if** $|p_t^X - p_t^Y|/p_t^Y > \alpha$ **then**

14: set $p_t^* = \omega p_t^X + (1-\omega)p_t^Y$

15: **end if**

Price smoothing.

16: Set $\bar{p}_t = \left(p_t^* + \sum_{k=t-L}^{t-1} p_k\right)/L + 1$.

Apply reserve price.

17: Use $\bar{p}_t$ as reserve price in periods $t, t+1, \ldots, t+M-1$.

Update posterior distribution (batch update).

18: Observe $\{\mathbb{I}\{\bar{p}_k \leq X_k\}, \max\{Y_k, \bar{p}_k\}\}_{k=t}^{t+M-1}$.

19: Update $\pi_{t-1}^X(\theta)$ using Equations (9) - (11) to get $\pi_{t+M-1}^X(\theta)$.

20: Update $\pi_{t-1}^Y(\theta)$ using Equations (10), (11) and (13) to get $\pi_{t+M-1}^Y(\theta)$.

21: Set $t = t + M$ and go to Line 7.

VI. Numerical Experiments

In this section we conduct experiments to evaluate the effectiveness of our proposed approach.

A. Dataset Description

In order to evaluate our method we use real-life data from ad auction markets. We use the publicly available iPinYou dataset [20], which contains information from the perspective of nine advertisers on a DSP. It contains information about bids placed by advertisers on a DSP for impressions during a week. For each bid there is information about the ad slot (height, visibility, etc.), time of day, the ad exchange, and the result of the bid. It is important to note that the dataset only contains information about the top bid and the second bid if the advertiser actually wins the auction. As a consequence the bid records represent a biased sample from the distribution of the top bid and second bid. However, the dataset contains information for several advertisers and could still be used to get a general understanding of the dynamics on the ad auction market. We use the iPinYou dataset to construct synthetic data for the top bid and second bid in order to test our proposed approach.

a) Construction of second bid: For a specific advertiser, we take the values of the second bid for the first 310000 impressions (sorted chronologically). We then divide these 310000 impressions into blocks with length 500. Within each block we sample with replacement 500 values of second bid from the 500 impressions in the block. After the sampling, we take a rolling mean with window length of 50 observations of the resulting time series. Finally, we take the last 300000 values of the resulting time series as the values for the second bid. The main reason for taking a rolling mean is that the advertisers in this dataset are bidding on ad slots from different publishers (with different properties etc.), whereas we are interested in a single publisher that is selling a specific ad slot. By taking a rolling mean we are effectively extracting the general trend in the second bids.

b) Construction of top bid: In order to construct the top bid, we take the time series of the second bid and divide the time series into blocks with length 500. Within each block we determine the maximum of the values of the second bid (denote the maximum by MB). The value of the top bid within a block is then equal to $MB \cdot (1+u)$ where $u \sim \mathcal{U}(0.0, 0.5)$. The draws of u are identically independently distributed (i.i.d) between blocks. This construction models a situation where the gap between the top bid and second bid varies over time and is independent of the level of the second bid.

We use data from 4 advertisers and we repeat the above procedure 5 times for each advertiser in order to generate 5 time series for the top bid and second bid.

B. Benchmark Strategies

The MAB framework is a popular framework for decision making under exploration-exploitation trade-offs. In order to judge the quality of our proposed method, we compare its performance with two MAB algorithms: (i) the UCB algorithm [21] and (ii) the EXP3 algorithm [22]. These are popular bandit algorithms that are simple to implement and have satisfactory performance in a broad range of applications. In the case of i.i.d and bounded rewards for each arm, UCB achieves an order-optimal upperbound on cumulative regret. In the adversarial setting with bounded rewards, EXP3 achieves a worst-case order-optimal upperbound on cumulative regret.

C. Settings and Performance Metrics

We consider two versions of our method: (i) TS-PF-RAP (model of Section V) and (ii) TS-PF-TB (model of Section IV). In order to measure the performance of the methods, we consider four performance metrics. The first performance metric is the cumulative average return, which is defined as $\sum_{t=1}^{T} \hat{R}_t/T$, where $\hat{R}_t$ is the observed return in period t. This is our main metric to determine the profitability of a strategy. The second metric is the success rate, which is defined as $\sum_{t=1}^{T} \mathbb{I}\{p_t \leq X_t\}/T$. This measures how often reserve prices are set too high. The third metric is the revenue rate, which is defined as $\sum_{t=1}^{T} \mathbb{I}\{p_t \leq X_t\}\hat{R}_t / \sum_{t=1}^{T} X_t$. This measures the rate at which the top bid is extracted. The fourth metric is the revenue rate given success, which is defined as $\sum_{t=1}^{T} \mathbb{I}\{p_t \leq X_t\}\hat{R}_t / \sum_{t=1}^{T} \mathbb{I}\{p_t \leq X_t\}X_t$. This measures the rate at which revenue is extracted given that a sale is successful. We average the three performance metrics over the 5 samples constructed for each advertiser.

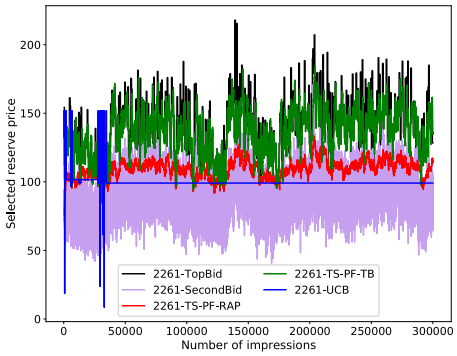
We run TS-PF-RAP with the following settings for the particle filter: $L = 50$, $M = 50$, $c = 0.5$, $\kappa = 0.015$, $K = 150$ particles. For the range of allowed prices we use $p_l = 1.0$, $p_h = 250.0$ since the resulting time series of the top bid is at most 250.0. The parameters for the prior distribution for the top bid are: $\theta_{0,k}^{(1)} \sim \mathcal{U}\left(\theta_{LB}^{(1)} = 1.0, \theta_{UB}^{(1)} = 250.0\right)$ and $\theta_{0,k}^{(2)} \sim \mathcal{U}\left(\theta_{LB}^{(2)} = 1.0, \theta_{UB}^{(2)} = 10.0\right)$. The parameters for the prior distribution for the second bid are identical.

The following settings are used for the risk-aware pricing component in TS-PF-RAP: (i) If $|p_t^X - p_t^Y|/p_t^Y \leq 0.1$ then set the reserve price according to $p_t^* = p_t^Y$; (ii) If $0.1 < |p_t^X - p_t^Y|/p_t^Y \leq 0.2$, then set the reserve price according to $p_t^* = 0.5p_t^X + 0.5p_t^Y$; (iii) If $|p_t^X - p_t^Y|/p_t^Y > 0.2$, then set the reserve price according to $p_t^* = 0.7p_t^X + 0.3p_t^Y$. This models a situation were the gap between X_t and Y_t can be "small", "medium" and "large", and for larger gaps p_t^* is closer to p_t^X. The settings for TS-PF-TB are identical to TS-PF-RAP except that there is no risk-aware pricing component. Instead of risk-aware pricing, TS-PF-TB optimizes Equation (4).

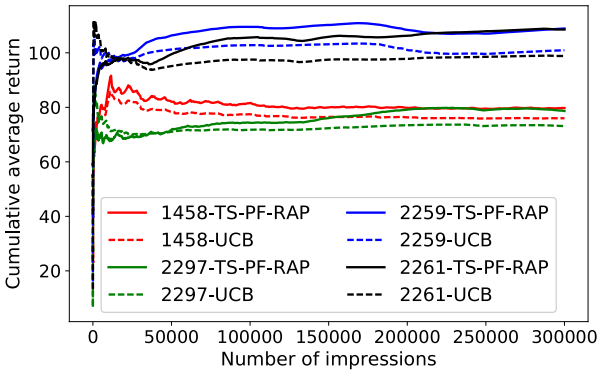
In the UCB and EXP3 algorithms each arm represents a reserve price and we use $N = 100$ arms which are equally spaced in the interval $[p_l, p_h]$.

D. Results: TS-PF-RAP versus bandits

The results for TS-PF-RAP, TS-PF-TB, UCB and EXP3 are displayed in Table I. The results show that TS-PF-RAP generally outperforms the other methods. An interesting observation is that UCB outperforms EXP3, even though EXP3 makes no assumption on the sequence of returns. A similar finding was also reported in [16]. As UCB outperforms EXP3, the rest of this subsection focuses on the differences between UCB and TS-PF-RAP. The performance gap differs depending on the specific advertiser, but in general, TS-PF-RAP has



(a) Reserve prices selected by TS-PF-RAP, TS-PF-TB and UCB for advertiser 2261.



(b) Results for advertisers 1458, 2297, 2259 and 2261.

Fig. 1. Cumulative average returns and selected reserve prices.

TABLE I
PERFORMANCE OF TS-PF-RAP, TS-PF-TB, UCB AND EXP3 ON SYNTHETHIC DATASETS.

	TS-PF-RAP			TS-PF-TB			UCB			EXP3		
advertiser 1458	min	max	mean	min	max	mean	min	max	mean	min	max	mean
cumulative average return	78.38	80.60	79.75	53.12	57.68	54.56	74.70	77.58	76.01	65.18	66.43	65.90
success rate	0.82	0.86	0.85	0.48	0.52	0.49	0.92	0.98	0.95	0.85	0.87	0.86
revenue rate	0.71	0.73	0.73	0.48	0.52	0.50	0.68	0.71	0.69	0.59	0.61	0.60
revenue rate success	0.81	0.83	0.82	0.90	0.91	0.90	0.69	0.75	0.71	0.67	0.70	0.69
advertiser 2297	min	max	mean	min	max	mean	min	max	mean	min	max	mean
cumulative average return	77.08	81.15	78.65	48.02	53.82	50.59	69.18	74.90	73.07	63.34	64.71	63.87
success rate	0.82	0.89	0.85	0.45	0.52	0.48	0.89	1.00	0.97	0.85	0.87	0.86
revenue rate	0.74	0.77	0.75	0.46	0.52	0.48	0.66	0.72	0.70	0.61	0.62	0.61
revenue rate success	0.85	0.87	0.86	0.92	0.93	0.92	0.67	0.78	0.72	0.70	0.70	0.70
advertiser 2259	min	max	mean	min	max	mean	min	max	mean	min	max	mean
cumulative average return	107.78	110.15	108.93	84.45	92.23	87.52	97.47	103.51	100.94	89.84	92.05	90.91
success rate	0.96	0.99	0.97	0.62	0.67	0.64	0.94	0.99	0.96	0.88	0.89	0.88
revenue rate	0.76	0.78	0.77	0.60	0.64	0.62	0.69	0.73	0.71	0.63	0.65	0.64
revenue rate success	0.77	0.81	0.78	0.91	0.91	0.91	0.69	0.76	0.73	0.70	0.72	0.71
advertiser 2261	min	max	mean	min	max	mean	min	max	mean	min	max	mean
cumulative average return	107.44	109.93	108.56	83.51	87.73	85.94	97.21	100.70	98.85	90.10	92.20	90.93
success rate	0.96	0.98	0.97	0.61	0.64	0.63	0.97	0.99	0.98	0.89	0.91	0.90
revenue rate	0.76	0.78	0.77	0.60	0.62	0.61	0.68	0.72	0.70	0.64	0.65	0.64
revenue rate success	0.77	0.80	0.78	0.91	0.92	0.91	0.69	0.73	0.71	0.70	0.72	0.71

a cumulative average return that is about 3.5 to 10.0 higher than UCB. The main explanation for the superior performance of TS-PF-RAP compared to UCB is that (i) TS-PF-RAP is able to better track changes in the top bid and (ii) TS-PF-RAP is better in selecting reserve prices that are closer to the top bid when the gap between the top bid and second bid is large. These two explanations are illustrated in Fig. 1a. This figure shows the selected reserve prices by TS-PF-RAP, TS-PF-TB and UCB relative to the top bid and second bid for a selection of advertisers (for a specific sample). From Fig. 1a we see that UCB tends to be more conservative and selects reserve prices that are often too low, which results in a higher success rate but a lower revenue rate (see also Table I). Fig. 1a shows that UCB generally does not react very quickly to changes in the top bid and second bid. On the other hand, the reserve prices selected by TS-PF-RAP are generally higher than those selected by UCB and they tend to do a better job at tracking the changes in the top bid and second bid. Finally, in Fig. 1b we can see that it does not take long for TS-PF-RAP to outperform UCB within the sales horizon.

E. Results: Impact of risk-aware pricing

If we compare TS-PF-RAP with TS-PF-TB, then we see that TS-PF-RAP significantly outperforms TS-PF-TB. The performance gap differs depending on the specific advertiser, but in general, TS-PF-RAP has a cumulative average return that is at least 15.0 higher than TS-PF-TB. TS-PF-TB tends to select reserve prices that are higher then those selected by TS-PF-RAP, and this results in a lower success rate and a higher revenue rate given success (see Table I).

Using Fig. 1a we can examine the performance of TS-PF-TB more closely. The figure shows that TS-PF-TB is able to track the top bid quite well, but if often sets reserve prices that are too high. This shows that estimation uncertainty of the distribution of the top bid can have a big impact on

performance. TS-PF-TB tracks the top bid well in terms of minimizing the mean square distance. However, in this application, exceeding the top bid by an amount $\Delta > 0$ is more costly than selecting a reserve price Δ below the top bid. This shows that simply tracking the top bid is not enough when the distribution of the top bid suffers from estimation errors.

Remark 4: We have presented results based on reasonable settings that performed well across all advertisers. The results for the other 5 advertisers in the iPinYou dataset are very similar to those reported in Table I and are omitted due to space limitations. Results with (i) $M = 25$ and $L = 100$, and (ii) block length of 100 in the construction of the top bid, are also similar. Performance can be improved by tuning the parameters for each advertiser separately. Also, the risk-aware pricing component can be refined to improve performance.

VII. Conclusion

We proposed a method for learning optimal reserve prices in second-price auctions. We studied a limited information setting where the values of the bids are not revealed and no historical information about the values of the bids is available. Our method is based on the principle of Thompson sampling combined with a particle filter to approximate and sample from the posterior distribution.

Our method can be improved in various ways. One direction for future work, is to make the risk-aware pricing component adaptive and self-regulatory. At the moment this component is fixed beforehand. Another interesting direction is to include features relating to the users in the model when setting reserve prices.

References

[1] J. Wang, W. Zhang, and S. Yuan, "Display advertising with real-time bidding (RTB) and behavioural targeting," *Foundations and Trends® in Information Retrieval*, vol. 11, no. 4-5, pp. 297–435, 2017. [Online]. Available: http://dx.doi.org/10.1561/1500000049

[2] S. Yuan, J. Wang, B. Chen, P. Mason, and S. Seljan, "An empirical study of reserve price optimisation in real-time bidding," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: ACM, 2014, pp. 1897–1906. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623357

[3] D. Austin, S. Seljan, J. Monello, and S. Tzeng, "Reserve price optimization at scale," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Oct 2016, pp. 528–536.

[4] P. Chahuara, N. Grislain, G. Jauvion, and J.-M. Renders, "Real-time optimization of web publisher RTB revenues," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: ACM, 2017, pp. 1743–1751.

[5] R. Refaei Afshar, Y. Zhang, M. Firat, and U. Kaymak, "A decision support method to increase the revenue of ad publishers in waterfall strategy," in *IEEE Conference on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, 2019, to appear.

[6] W. Wu, M.-Y. Yeh, and M.-S. Chen, "Deep censored learning of the winning price in the real time bidding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 2526–2535. [Online]. Available: http://doi.acm.org/10.1145/3219819.3220066

[7] W. Zhu, W. Shih, Y. Lee, W. Peng, and J. Huang, "A gamma-based regression for winning price estimation in real-time bidding advertising," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec 2017, pp. 1610–1619.

[8] W. C.-H. Wu, M.-Y. Yeh, and M.-S. Chen, "Predicting winning price in real time bidding with censored data," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 1305–1314. [Online]. Available: http://doi.acm.org/10.1145/2783258.2783276

[9] N. Cesa-Bianchi, C. Gentile, and Y. Mansour, "Regret minimization for reserve prices in second-price auctions," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 549–564, Jan 2015.

[10] Z. Xie, K.-C. Lee, and L. Wang, "Optimal reserve price for online ads trading based on inventory identification," in *Proceedings of the ADKDD'17*, ser. ADKDD'17. New York, NY, USA: ACM, 2017, pp. 6:1–6:7. [Online]. Available: http://doi.acm.org/10.1145/3124749.3124760

[11] M. R. Rudolph, J. G. Ellis, and D. M. Blei, "Objective variables for probabilistic revenue maximization in second-price auctions with reserve," in *Proceedings of the 25th International Conference on World Wide Web*, ser. WWW '16. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, pp. 1113–1122. [Online]. Available: https://doi.org/10.1145/2872427.2883051

[12] M. Mohri and A. M. n. Medina, "Learning algorithms for second-price auctions with reserve," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2632–2656, Jan. 2016. [Online]. Available: http://dl.acm.org/citation.cfm?id=2946645.3007027

[13] E. Ikonomovska, S. Jafarpour, and A. Dasdan, "Real-time bid prediction using Thompson sampling-based expert selection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 1869–1878. [Online]. Available: http://doi.acm.org/10.1145/2783258.2788586

[14] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, ser. NIPS'11. USA: Curran Associates Inc., 2011, pp. 2249–2257. [Online]. Available: http://dl.acm.org/citation.cfm?id=2986459.2986710

[15] A. Nuara, F. Trovò, N. Gatti, and M. Restelli, "A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns," 2018. [Online]. Available: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16819

[16] G. Jauvion, N. Grislain, P. Dkengne Sielenou, A. Garivier, and S. Gerchinovitz, "Optimization of a SSP's header bidding strategy using Thompson sampling," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 425–432. [Online]. Available: http://doi.acm.org/10.1145/3219819.3219917

[17] D. J. Russo, B. V. Roy, A. Kazerouni, I. Osband, and Z. Wen, "A tutorial on Thompson sampling," *Foundations and Trends® in Machine Learning*, vol. 11, no. 1, pp. 1–96, 2018. [Online]. Available: http://dx.doi.org/10.1561/2200000070

[18] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models (Springer Series in Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2005.

[19] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *Trans. Sig. Proc.*, vol. 50, no. 3, pp. 736–746, Mar. 2002. [Online]. Available: http://dx.doi.org/10.1109/78.984773

[20] W. Zhang, S. Yuan, J. Wang, and X. Shen, "Real-time bidding benchmarking with iPinYou dataset," *arXiv preprint arXiv:1407.7073*, 2014.

[21] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2, pp. 235–256, May 2002. [Online]. Available: https://doi.org/10.1023/A:1013689704352

[22] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002. [Online]. Available: https://doi.org/10.1137/S0097539701398375