European Journal of Control 30 (2016) 31–49

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Control problems in online advertising and benefits of randomized bidding strategies

Niklas Karlsson

AOL Platforms R&D, 395 Page Mill Road, 3rd Floor, Palo Alto, CA 94306, USA

ARTICLE INFO

Article history:
Received 30 January 2016
Accepted 20 April 2016
Recommended by A. Astolfi
Available online 9 May 2016

Keywords:
Advertising
Optimization
Modeling
Bidding
Randomization
Exploration and exploitation

ABSTRACT

Online advertising is a US$600 billion industry where feedback control has come to play a critical role. The control problems are challenging and involve nonlinearities including discontinuities, high dimensionality, uncertainties, non-Gaussian noise, and more. In this paper systems engineering principles are applied to a core optimization problem within online advertising. First we demonstrate how the optimization problem may be decomposed into separate low-level estimation and high-level control modules. Then we derive a plant model from first principle to show how uncertainties and noise propagate through the plant. The plant model reveals challenges of the control problem and provides a framework to assess the impact on the plant behavior from different designs of the low-level estimation module. Thereafter, we describe a bid randomization technique that can be used in various ways to improve the performance and robustness of the system. The bid randomization technique is finally used to develop an algorithm for exploration and exploitation of an auction-based network, furnishing a solution to the above estimation subproblem.

1. Introduction

1.1. Overview

Over the past decade, the online advertising industry has grown dramatically in size, significance, and complexity; and an ever-growing number of companies have created a business model around some aspect of advertising. For some of them the goal is to use data-driven automation and optimization to manage the marketing budgets of advertisers. The optimization involves constraints imposed by the advertisers, as well as dynamics, uncertainties, and noise resulting from interactions across ad campaigns and between ad campaigns and Internet users. For that reason it is not surprising that feedback control plays a critical role in online advertising.

The competition among the optimization providers is fierce, the number of advertising campaigns they manage is large, and the amount of data being collected about user behaviors and utilized in the decision making process is massive, making the optimization and control problem high-dimensional and complex. Indeed, the complicated dynamics include time-varying non-linearities (to the point of discontinuities) and involve non-Gaussian disturbances and unknown latency distributions.

Advertising online is different in many ways from advertising in traditional media, such as printed magazines, but the fundamental problems facing the advertising professional are the same. Ultimately, all advertising is about showing the right ad to the right consumer at the right time. The difference between online and traditional advertising is that the data available to make fact-based decisions and the immediate publishing of information and content on the Internet is not limited by geography or time. Tractable (yet challenging) feedback control problems are formed by carefully establishing objective functions, defining measurement and control signals, and designing a system with a sufficiently high sample rate and short delays.

According to eMarketer, online advertisers worldwide spent approximately $600 billion in 2015 [11]. This already large number is expected to grow, but as the problem statements and solutions for online advertising are largely the same as those for online content distribution and e-commerce (YouTube, Netflix, Amazon, and so forth), the business opportunity of online optimization and control systems is much larger. Given the current size of the online advertising industry and its rapid growth, there are strong reasons to invest in automated algorithmic solutions to the problem of serving ads.

At the advent of online advertising (pre-2005), it was popular to pose the optimization problem as a static linear network optimization problem subject to campaign constraints [12]. This was made possible by implementing a stand-alone prediction system for all relevant stochastic quantities and relying on the certainty

E-mail address: Niklas.Karlsson@teamaol.com

http://dx.doi.org/10.1016/j.ejcon.2016.04.007
0947-3580/

equivalence principle to establish a *Linear Program* (LP) problem, which was solved using a standard centralized LP solver. Over time, however, this centralized approach began to break down due to the growth of the networks and the addition of business logic that did not fit into the linear program formulation.

The second generation of optimization algorithms (2005–2012) involved decentralization to support greater scale and incorporated feedback control to provide an ability for the system to learn from its mistakes and handle dynamic environments [23,20]. However, the optimization was still typically implemented in a closed network, optimizing towards a single network-centric objective.

In recent years (2012–) the number of participants in various aspects of online advertising has continued to grow, and advertisers have become more savvy. The increasing number of participants has led to the development of open auction exchanges where ad views are traded by way of bidding. An open exchange for online advertising is also referred to as a *Real Time Bidding* (RTB) exchange. With more savvy advertisers, the providers of optimization technologies have been forced to design optimization algorithms that are advertiser-centric in the sense that bids submitted on behalf of an advertiser maximize an objective function that is aligned with the objectives of this advertiser. This is the state of the industry today, and the focus now (in 2016) is to enhance the optimization algorithms for improved performance, simultaneous optimization across multiple platforms (desktop, mobile, video, TV), and support of multiple campaign objectives.

To prepare for the main content of the paper, let us introduce some key concepts of RTB based online advertising and of the related optimization. In this context there are Internet users, publishers, advertisers, and one or more exchanges where ad inventory is traded between publishers and advertisers. Publishers may set a reserve price, which is the lowest acceptable price at which they are willing to sell ad space on their web sites. Advertisers on the other hand may submit a bid for the opportunity to show an ad. Typically there are also agents representing the publishers and advertisers to provide intelligent reserve price and bidding strategies that aim to maximize the returned value for each principal. Here we do not distinguish between agent and principal but instead say e.g. “the advertiser” or “the campaign” submits a bid, when it is really an agent of the advertiser doing it.

Publishers own web pages and receive traffic from Internet users. The traffic is monetized by selling advertisement space to advertisers. The delivery of an online ad campaign involves impressions, where an *impression* is one view of an ad. Technically, an impression may be in-view or out-of-view, but in this paper we do not make such a distinction. An impression by itself carries a branding value since an Internet user's exposure to the ad leads to brand recognition, which in the future may lead to (untraceable) sales of the brand on- or off-line. An impression also carries a performance value in that it potentially leads to user-initiated traceable actions, ranging from a simple click on the ad to the completion of an online survey or the placement of an order for a magazine subscription, an insurance plan, a vacation trip, and so forth.

Typically the impression allocation for RTB based advertising is governed by a sealed second price standard auction [25]. This allocation mechanism is assumed throughout this paper. The process begins with an Internet user loading a web page containing some advertisement space. Immediately, the publisher of the web page sends a request for an ad to an impression exchange. The ad impression request contains information about the user to help advertisers estimate the value of showing an ad to this particular user at this specific time. The information provided depends on the privacy settings of the user, the device on which the user loads the web page (e.g. desktop or mobile), the web browser that is being used, and so on; but the advertisers can use this information to varying degrees of confidence to infer the user's location, age, gender, interests, etc. In the open exchange any interested campaign may submit a bid for the opportunity of showing an ad to the Internet user. The campaign submitting the highest bid is awarded the impression and is charged an amount equal to the second highest bid. In reality, an additional fee such as a commission and/or data fee is added to this cost, but it is neglected in this paper. The winning bidder's ad is finally served to the user. Cooperation among bidders is not permitted, and the industry standard on response time requirement is 50 ms from when the impression request is sent to the exchange to when the ad is served to the user.

The total cost to the advertiser is given by the cumulative cost of all awarded impressions, and the cost for an individual impression is determined by the second highest bid price. The total value, on the other hand, is given by the total number of events of interest times the monetary value the advertiser attributes to each event, where we use *event* as the general term denoting any one of impression, click, or conversion that the advertiser is interested in. Note, in this paper we assume that an advertiser is interested in only one type of event and assigns the same monetary value to each such event. The multi-objective problem and the problem where events may have different values are of relevance, but require a more elaborate presentation.

The overarching goal of ad campaign optimization is to maximize the total generated monetary value of the campaign while maintaining an even delivery of the advertising budget and a minimum required *Return on Investment* (ROI), where ROI for a campaign is defined as the ratio between total monetary value and total cost. A campaign flight is most often one month long, but may be as short as a few hours or as long as many months. The campaign budget often takes a value in the wide range between $100 and $100,000 per day, but sometimes takes an even more extreme value outside of this range.

For a pure branding campaign the optimization goal translates into serving as many ad impressions as possible without violating the total cost or ROI constraints, which simply means to buy the least expensive impressions on the exchange. For a performance campaign defined by some conversion event, the goal is to buy impressions such that the maximum number of conversions is generated without violating the constraints on total cost or ROI.

1.2. Literature review and contributions

There is a huge body of literature on auction theory [25], decision and game theory in general [7,28,13], and dynamic game theory in particular [6]; and papers have been written about applications in online advertising, but in the prior work plant dynamics is typically either ignored or is unrealistic and assumed given. Whenever feedback control is part of the solution it is critical to consider the plant dynamics, and besides vague descriptions of the dynamics in some literature [20], the current paper is, to our best knowledge, the first attempt at systematically describing the input–output behavior of an auction-based online advertising optimization system, where the input includes a control signal as well as various noise signals. The modeling is qualitative in nature, but is based on first principle reasoning and provides important insights helpful in the design of both estimation and control algorithms.

Challenges from doing simultaneous estimation and control and the necessary coupling among multiple bidding agents in an auction-based system made us propose a bid randomization mechanism [17] with a range of novel use cases (e.g. [19,18]) to improve, and sometimes enable, estimation and control of an auction-based network. An important reason why the proposed

bid randomization mechanism helps is that in a setting with a very large number of auctions, such as in online advertising where each auction relates to an impression request, it effectively removes discontinuities. The sharp impact of gradually changing bid prices is replaced by a gradual impact, and the mathematical analysis can leverage tools from smooth dynamical systems.

One use case in particular relates to the so-called *exploration and exploitation* or *multi-armed bandits* problem, which takes its name from a traditional slot machine (one-armed bandit). When pulled, each lever provides a reward drawn from a distribution associated with that specific lever. The objective of the gambler is to maximize the sum of rewards earned through a sequence of lever pulls. A celebrated early solution of the multi-armed bandit problem involves Gittins index [15]. It is a measure of the potential reward as opposed to expected reward and systematically balances between exploration and exploitation. However, calculating Gittins' index is, in general, NP-complex and is based on strict assumptions such as infinite time horizon, zero measurement delay, and/or time invariance. Other approaches towards solving multi-armed bandit problems include various greedy strategies [4,5,14,24,26,27]. While these sometimes can be made efficient, they are normally non-adaptive and difficult to tune.

In recent years a randomization-based technique under the name of Thompson sampling has received renewed attention. It was originally proposed in 1933 [29] as an algorithm that is simple to implement and has good empirical performance, but the algorithm did not become mainstream probably because it lacked a solid theoretical justification. The algorithm also goes under the name posterior-sampling, which is more descriptive since it basically produces each bid from the posterior distribution of the optimal bid (in Bayesian sense).

We independently proposed an exploration and exploitation algorithm [18] that resembles Thompson sampling, but with a memory loss aspect that leads to a fixed point solution, adaptivity, and the capability of shaping the transient and steady-state behavior of the algorithm. In this paper we suggest how to use techniques from nonlinear dynamics to analyze the behavior of the algorithm for improved performance and more insights. The algorithm is implemented and branded as ContentLearn™ [21] by AOL. It provides enhanced exploration and exploitation, and improves the prospects of successfully augmenting a feedback controller to the system.

Over the last few years several important theoretical results have been derived for Thompson sampling [29,1,2,8,16]. The results primarily deal with the expected asymptotic regret, where *regret* refers to how much higher the expected reward would be if we knew beforehand which lever to pull. In the above references it has recently been proven that Thompson sampling has excellent asymptotic properties, which together with the simplicity of implementing the algorithm makes it a very compelling algorithm to use in applications. However, Thompson sampling as proposed in the literature does not strictly incorporate adaptivity. Moreover, the above work omits a dynamical analysis of the algorithm.

1.3. Key notation

Throughout this paper index k denotes a time stamp and $k+1$ denotes the time instant following k. We use subscript i to denote a specific *segment* of impressions assumed to be indistinguishable to the advertisers. A segment may represent a type of users, e.g. "Female Users between 20 and 30 years of Age," "Tech Geeks," or "Internet Users that have not seen a specific ad before," but may as well be defined as "Any User" or "A Unique User". If time or impression in some context is irrelevant, then we may drop k or i from the equations for ease of reading.

Throughout the paper we are often interested in how a multivariate function depends on a scalar control signal u. For example, a function may be given by $c(b_1, a_1, b_2, a_2, \ldots)$, where $b_i = up_i$ and $a_i = 1$ for all i. The notation $c(u)$ in this paper is then interpreted as the function $c(up_1, a_1, up_2, a_2, \ldots)$ where $p_1, p_2, \ldots$ are held constant and $a_1 = a_2 = \cdots = 1$.

A capital letter typically represents a random variable while the letter in lower case represents an observed realization. Moreover, we use $\sim$ to denote "distributed as" in derivations related to random variables. For example, we may say $N_E \sim \text{Binomial}(n_I, p)$ to represent the statement that N_E is a binomial random variable with parameters n_I and p.

2. Control problems in online advertising

2.1. Overarching problem formulation

The impression allocation for segment i is governed by a sealed second price auction, where b_i is the bid price used in the auction and a_i is the bid allocation, or the sampled fraction of auctions we choose to participate in. We refer to the latter as *fractional bidding*. The highest competing bid price is denoted $b_i^\star$, and the available number of impressions $n_{I,i}^{tot}$. We assume that an advertiser is interested in only one type of *event*, e.g. an impression, click, or conversion, and attaches a monetary value of v_E to each such event. An event is conditioned on an impression first taking place and the probability of an event to occur is then p_i. Note, if the event of interest to the advertiser is an impression, then $p_i \equiv 1$.

The expected total number of impressions from segment i awarded to the campaign is $n_{I,i} = a_i \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot}$, where for simplicity we assume that the auction is always won if $b_i \geq b_i^\star$ and where $\mathbb{I}_X$ is the indicator function satisfying $\mathbb{I}_X = 1$, if $X = \text{true}$, and $\mathbb{I}_X = 0$, if $X = \text{false}$.

By definition of a second price auction, the advertiser pays the highest competing bid for any awarded impression, hence the total cost of the above impressions is $c_i = b_i^\star a_i \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot}$. Since the advertiser attributes a value v_E to events that occur with a probability of p_i given an impression has been awarded, the effective value of one impression is $p_i v_E$, which for all awarded segment i impressions corresponds to $v_i = v_E p_i a_i \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot}$. The campaign-level impression volume n_I, cost c, and expected value v are obtained by summing across all segments targeted by the campaign; and the expected *Return on Investment* (ROI) is defined by $r = v/c$. Hence,

$$n_I = \sum_i a_i \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot} \tag{1}$$

$$c = \sum_i a_i b_i^\star \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot} \tag{2}$$

$$v = \sum_i a_i p_i v_E \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot} \tag{3}$$

$$r = \frac{\sum_i a_i p_i v_E \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot}}{\sum_i a_i b_i^\star \mathbb{I}_{\{b_i \geq b_i^\star\}} n_{I,i}^{tot}}. \tag{4}$$

The overarching objective is to devise a bidding strategy b_i, a_i, for $i = 1, 2, \ldots$, that maximizes the expected total advertising value v without violating constraints on cost c and ROI r; i.e., the optimization objective is to

$$\max_{\substack{b_i \geq 0 \\ 0 \leq a_i \leq 1 \\ i = 1,2,\ldots}} v(b_1, a_1, b_2, a_2, \ldots) \tag{5}$$

subject to

$$c(b_1, a_1, b_2, a_2, \ldots) \le u_{cost}^{ref} \tag{6}$$

$$r(b_1, a_1, b_2, a_2, \ldots) \ge u_{ROI}^{ref}, \tag{7}$$

where u_{cost}^{ref} and u_{ROI}^{ref} are constraints prescribed by the advertiser. A natural choice is to set $u_{ROI}^{ref} = 1$ since it implies that the advertiser is interested in running the campaign as long as the value is larger than the cost; however, if the delivery of the campaign is also subject to, e.g., a 1% commission fee on the accrued cost, then a more sensible choice is $u_{ROI}^{ref} = 1.01$.

It can be shown (see Appendix A) that an optimal bidding strategy is given by the following bid price and bid allocation: The optimal bid price is $b_i = u^{opt} p_i$, where a campaign-level variable $u^{opt} \ge 0$ is assigned the smallest value possible for which at least one of the constraints is binding or violated (if a_i were to equal one for all i). The optimal bid allocation a_i equals one except for in the segment containing the lowest ROI awarded impressions. This segment is denoted i'. If constraint violation cannot happen for any values of b_i, a_i, $i = 1, 2, \ldots$, then u^{opt} and a_i are selected so that all impressions are awarded, which is achieved if $u^{opt} = \max_{p_i > 0} b_i^\star / p_i$ (assuming $p_i > 0$ for at least one i).

In other words, the optimal bidding strategy is

$$b_i = u^{opt} p_i, \quad i = 1, 2, \ldots \tag{8}$$

where the campaign-level factor u^{opt} is given by

$$u^{opt} = \min_u \left\{ u \left| \sum_i b_i^\star \mathbb{I}_{\{u p_i \ge b_i^\star\}} n_{I,i}^{tot} \ge u_{cost}^{ref} \text{ or } \sum_i p_i v_E \mathbb{I}_{\{u p_i \ge b_i^\star\}} n_{I,i}^{tot} \le u_{ROI}^{ref} \sum_i b_i^\star \mathbb{I}_{\{u p_i \ge b_i^\star\}} n_{I,i}^{tot} \text{ or } u = \max_{p_i > 0} \frac{b_i^\star}{p_i} \right. \right\}, \tag{9}$$

and where the segment-level bid allocation a_i is

$$a_i = \begin{cases} \min\left(\dfrac{u_{cost}^{ref} - \sum_{i=1}^{i'-1} b_i^\star n_{I,i}^{tot}}{b_{i'}^\star n_{I,i'}^{tot}}, \dfrac{\sum_{i=1}^{i'-1} \left(u_{ROI}^{ref} b_i^\star - v_E p_i \right) n_{I,i}^{tot}}{\left(p_{i'} v_E - u_{ROI}^{ref} b_{i'}^\star \right) n_{I,i'}^{tot}} \right), & \text{if } i = i' \\ 1, & \text{otherwise.} \end{cases} \tag{10}$$

Segment i' is the value of i for which $u^{opt} p_i = b_i^\star$, which we for simplicity assume is unique. Also, if there is no such i', then $a_i = 1$ for all i.

To better understand the bidding strategy, note that with a strategy $b_i = u p_i$ an impression is awarded for a small value of u only if p_i is *large* or $b_i^\star$ is *small* (an impression is awarded if $u p_i \ge b_i^\star$). Since the ROI of impressions from segment i equals $v_E p_i / b_i^\star$, only high ROI impressions are awarded for small u. The optimal strategy is to use the smallest possible u to reach or exceed one of the constraints; i.e., impressions with high expected ROI are favored and impressions with low ROI are considered only as needed to deliver the ad budget, but only for as long as we do not violate the cost (budget) and ROI constraints. If the available impressions in segment i' would result in a constraint violation, then we only bid for a fraction of those lowest ROI impressions to prevent a violation of the constraints. Keep in mind, we outbid the competition in all segments where $u^{opt} p_i \ge b_i^\star$, and i' is the segment among these for which the ROI $v_E p_i / b_i^\star$ is the smallest.

If fractional bidding is not used ($a_i = 1$ for all i), then the best possible bidding strategy is given by $b_i = u^{opt} p_i$, where u^{opt}, by necessity, is the largest value for which neither of the constraints is violated. This strategy is inferior to the strategy including the fractional bidding since no impression (and no value) is generated from segment i'.

Fractional bidding can be disregarded if the maximum possible contribution from segment i' is very small; i.e., if the optimal $a_{i'} \approx 0$. If we cannot disregard the contribution from segment i', then fractional bidding is required for optimality. Unfortunately, computing the optimal $a_{i'}$ using (10) is not practical — it is computationally expensive and numerically sensitive since it requires accurate estimates of many uncertain segment-level quantities (already the determination of i' is difficult).

We introduce an alternative to fractional bidding in Section 3. This alternative does not require segment-level information and has additional benefits in terms of enhancing the robustness of the network. For the reminder of this section, however, we do not use fractional bidding in any shape or form; i.e., assume $a_i = 1$ for all i which simplifies the analysis of the bidding strategy.

Without the added complexity of fractional bidding, the optimal bidding strategy may be decomposed into two components. In the first component, segment-level event rates p_i are provided independently of the campaign-level constraints. Separately, in the second component, a campaign-level factor u is calculated based exclusively on campaign-level information to maximize the campaign-level monetary value while satisfying the constraints.

Of great importance as we soon switch perspective and think of the problem as a feedback control problem, is that with a bidding strategy $b_i = u p_i$, the cost $c(u)$, value $v(u)$, and ROI $r(u)$ can be shown to be monotonic functions of u (see Appendix B). Indeed, given fixed values of a_i, p_i for all i; $c(u)$ and $v(u)$ are non-decreasing functions while $r(u)$ is a non-increasing function.

The above optimal bidding strategy and the monotonicity properties of $c(u)$, $v(u)$, and $r(u)$ is the basis of the control problem analyzed in this paper. However, instead of using (9) to calculate u^{opt}, the optimal value of u is obtained by way of feedback control. In principle, the feedback signal consists of $c(u)$ and $r(u)$, while the reference signal consists of u_{ROI}^{ref} and u_{ROI}^{ref}. The control objective turns into a reference tracking problem where u is adjusted in such a way that $c(u) \approx u_{cost}^{ref}$ or $r(u) \approx u_{ROI}^{ref}$, depending on which constraint is being violated at the smallest value of u.

However, the event rates p_i's in practice are unknown and may change over time, hence must be estimated adaptively. Consequently, the estimates are subject to errors that may impact the optimization result (including the performance and stability of the control system). Similarly, the competitive landscape dictated by the $b_i^\star$'s typically also changes over time altering what are high and low ROI segments. Finally, the available impression volumes $n_{I,i}^{tot}$'s also vary over time influencing what is the optimal value u^{opt}.

Introduce time index k in all previously defined quantities and denote the estimated event rate $\hat{p}_i(k)$. Suppose furthermore all involved variables remain constant within one time interval defined by k. If the estimate $\hat{p}_i(k)$ is reasonably accurate, then we assume that an approximately optimal bidding strategy is given by:

$$b_i(k) = u(k) \hat{p}_i(k).$$

Supported by the certainty equivalence principle under the premise of $\hat{p}_i(k) \approx p_i(k)$, we now pose the problem of computing $u(k)$ as a feedback control problem where all $\hat{p}_i(k)$'s are part of a perceived plant and where the difference $\hat{p}_i(k) - p_i(k)$ is treated as process noise or plant uncertainty.

A high-level representation of the control system is shown in Fig. 1. The plant is defined by the mapping from control signal $u(k)$ to plant output $y(k) = [y_{value}(k), y_{cost}(k)]^T$, which represents measured campaign-level monetary value and cost. Note, $v(k)$ and $c(k)$ are subject to measurement noise and latency (discussed further in Section 2.2). The plant dynamics can be described by a general discrete-time difference equation

$$x(k+1) = f(k, x(k), u(k), w_1(k))$$
$$y(k) = g(k, x(k), u(k), w_2(k)),$$

where the state x, the state equation $f(\cdot)$, and the output equation $g(\cdot)$, as well as the distributions of process and measurement

noises w_1 and w_2 are determined by systematic modeling. The sample time of the discrete-time system is application-dependent, but for an advertising optimization system it may be between e.g. 5 and 15 min.

The objective is to design a feedback control algorithm from plant output $y(k)$ and reference input $u^{ref}(k) = [u^{ref}_{cost}(k), u^{ref}_{ROI}(k)]^T$ to control signal $u(k)$, which maximizes the estimated value $\hat{v}(u(k))$ without violating any of the two constraints. For scalability reasons, the feedback controller has no explicit knowledge of true or estimated event rates p_i and $\hat{p}_i(k)$, highest competing bid prices $b^\star_i$, available impression volumes $n^{tot}_{I,i}(k)$, or latency distributions.

As a practical side note, advertisers typically permit distributing the delivery of the campaign budget throughout each day in a way that reflects the time-of-day pattern in available impression supply volume. This is of business importance since it means we can avoid buying a disproportional number of impressions during hours of the day when the impression supply (and Internet traffic) is low. It is also important from a control design perspective since it makes it possible to do estimation and control of the plant at approximately the same operating point of u throughout the day rather than competing at very different price points at different times of day, with potentially very different properties of the plant in terms of plant gain, and so forth. For the above reasons, the constraint on the cost $c(k)$ is typically relaxed to hold only as a 24 h moving average.

This paper does not solve the control problem, but derives important properties of the plant (Section 2.2), identifies challenges a feedback controller must deal with (Section 2.3), and proposes a versatile bid randomization technique (Section 3) that can be used in various ways to mitigate some of the challenges (e.g. Section 4).

2.2. System modeling

In this section we investigate the mapping from plant input $u(k)$ to plant output $y(k) = [y_{value}(k), y_{cost}(k)]^T$ in Fig. 1. The goal is to develop a qualitative appreciation of the plant behavior, an understanding of how uncertainties and noise propagate through the plant, and a recognition of how the plant behavior may vary from one campaign to the next. This supports a separate undertaking to design control algorithms for advertising campaigns. It also helps the account manager charged to configure campaigns (deciding what segments the campaign is eligible for), or the engineer tasked to design algorithms to estimate p_i to understand the system-level impact of their choices.

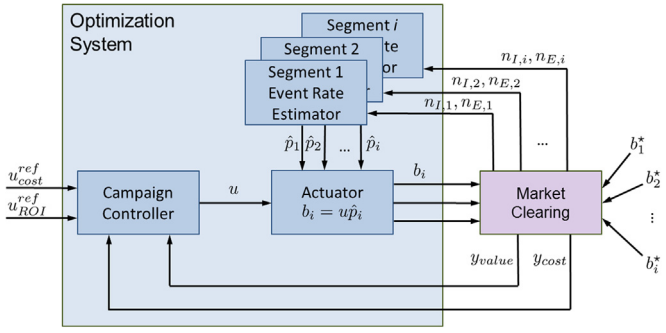

Fig. 1. High-level block diagram of the control system used for optimization of online advertising.

Let us begin by analyzing online advertising from first principle. From this perspective, we argue segments can be modeled one at a time, and described using five subsystems connected as shown in Fig. 2, where the outputs of each segment system can be summed up for a campaign-level output according to $y_{value}(k) = \sum_i y_{value,i}(k)$ and $y_{cost}(k) = \sum_i y_{cost,i}(k)$. Take note of the causal relationship among signals: Control signal $u(k)$ is needed to produce a bid $b_i(k)$, which is needed for impressions $n_{I,i}(k)$ to be awarded and a cost $c_i(k)$ to incur. Furthermore, impressions are needed to generate value-bearing events $n_{E,i}(k)$ and advertising value $v_i(k)$, and finally the advertising value and cost are needed before they can be measured as $y_{value,i}(k)$ and $y_{cost,i}(k)$.

Bid Generation represents the mapping from control signal $u(k)$ to bid price $b_i(k)$, and is governed by $b_i(k) = u(k)\hat{p}_i(k)$ plus actuator delay. Note, $u(k)$ is typically computed before the impression request from the publisher is received and is not subject to the 50 ms response time requirement mentioned in Section 1. Nevertheless, the actuator delay is relatively short and typically in the order of a few minutes, which is insignificant relative to other dynamics in the plant and controller. We can therefore safely disregard this delay. Of greater importance is how $\hat{p}_i(k)$ impacts the bidding. The true event rate $p_i(k)$ is usually slowly varying, and is, for the sake of modeling, assumed constant; however, in the field of online advertising p_i is typically very small and oftentimes in the range of 10^{-6} to 10^{-4}. Meanwhile the historical time series data of $n_{I,i}(k)$ and $n_{E,i}(k)$ needed to compute $\hat{p}_i(k)$ is limited making the estimation a challenge. Indeed, the variance of $\hat{p}_i(k)$ often ends up being large, say, 10% of the true event rate p_i (sometime much larger) and the distribution of $\hat{p}_i(k)$ asymmetric (non-Gaussian), especially in the early phase of a campaign before a sufficient amount of historical impression and event data for a specific segment is available.

Making things worse, before historical data is available, many algorithms designed to compute $\hat{p}_i(k)$ for use in an auction environment purposely add a large positive bias to the estimate to ensure that $b_i(k)$ is large enough to win essential impressions required for learning. The balancing act between using the best unbiased estimate for the most appropriate valuation of the next impression, versus boosting the estimate to win more impressions and improve the event rate estimate and thereby the valuation of future impressions is in the literature referred to as the exploration and exploitation trade-off. Exploration and exploitation is challenging in its own right because of system delays and computational complexity, but the design choice made for this algorithm also has tremendous impact on the feedback control system and effectively adds what can be viewed as significant process noise and/or plant uncertainty. In particular, it may result in $r(u)$ not being a monotone function of u, effectively introducing positive feedback into the feedback control loop.

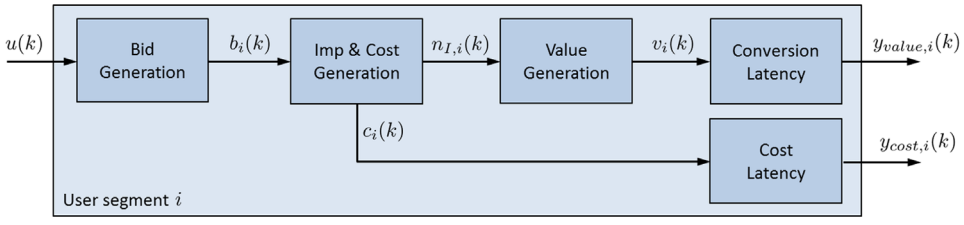

Fig. 2. The plant decomposed into five subsystems consisting of a cascade of Bid Generation, Impression and Cost Generation, Value Generation, and Conversion Latency, and a branch consisting of Cost Latency.

We represent the event rate estimate using a multiplicative noise model $\hat{p}_i(k) = p_i(1+\epsilon_{\hat{p},i}(k))$, where $\epsilon_{\hat{p},i}(k)$ is a stochastic process satisfying $-1 \le \epsilon_{\hat{p},i}(k) \le -1+1/p_i$ and with properties such as mean and auto-covariance heavily dependent on the specific algorithm used for the estimation of $\hat{p}_i(k)$. The multiplicative structure makes it plausible as an approximation that the same noise model can be used for many or all segments i, even if p_i differs by orders of magnitude. In other words, let

$$b_i(k) = p_i(1+\epsilon_{\hat{p},i}(k))u(k). \tag{11}$$

If $\hat{p}_i(k)$ is computed separately for each segment and based on time-series data, then we expect $\epsilon_{\hat{p},i}(k)$ to be approximately uncorrelated across segments (while correlated over time).

Impression and Cost Generation represents the mapping from bid price $b_i(k)$ to awarded impression volume $n_{I,i}(k)$ and cost $c_i(k)$. The second price auction dictates that the highest bidder wins an impression and pays an amount equal to the second highest bid for the impression, hence, $n_{I,i}(k) = \mathbb{I}_{\{b_i(k) \ge b_i^\star(k)\}} n_{I,i}^{tot}(k)$ and $c_i(k) = b_i^\star(k)\mathbb{I}_{\{b_i(k) \ge b_i^\star(k)\}} n_{I,i}^{tot}(k)$. However, $b_i^\star(k)$ and $n_{I,i}^{tot}(k)$ are volatile quantities as a result of dynamic and imperfect bidding from competitors and an ever-changing behavior of Internet users. We propose using a highest competing bidder model $b_i^\star(k) = \overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)$, where $\overline{b}_i^\star$ is approximately constant and $\epsilon_{b^\star,i}(k)$ is a stochastic process satisfying $\epsilon_{b^\star,i}(k) \ge -1$ and with properties such as mean and auto-covariance heavily dependent on the competitive landscape, including the density of bids and the specific algorithms used to compute competing bids. Note, $b_i^\star(k)$ is an order statistic [9] since it is the maximum bid among all competing bidders — an important fact to consider during the modeling of its dynamic and static properties.

For guidance to select a model for the impression volume, inspect Fig. 3 which shows a representative example of $n_{I,i}^{tot}(k)$ versus k. Note the significant time-of-day pattern and the standard deviation that appears approximately proportional to the impression volume. With this in mind, we propose a traffic model $n_{I,i}^{tot}(k) = h_{seas}(k)\overline{n}_{I,i}^{tot}(1+\epsilon_{I,i}(k))$, where $h_{seas}(k)$ is a 24 h periodic deterministic function, $\overline{n}_{I,i}^{tot}$ is approximately constant, and $\epsilon_{I,i}(k)$ is a stochastic process with mean zero and variance σ_I^2 satisfying $\epsilon_{I,i}(k) \ge -1$. The multiplicative noise structure is again useful since

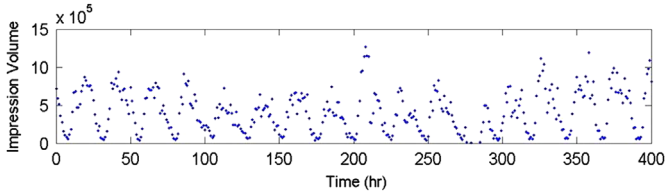

Fig. 3. Representative example of available impression volume $n_{I,i}^{tot}(k)$ versus time k.

$b_i^\star(k)$ and $n_{I,i}^{tot}(k)$ tend to have approximately scale-invariant standard deviations. Note, $h_{seas}(k)$ is modeled as segment-independent, which is a reasonably accurate model for any campaign targeting users in at most a few time zones in close proximity to one another. In conclusion, we obtain

$$n_{I,i}(k) = \mathbb{I}_{\left\{b_i(k) \ge \overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)\right\}} h_{seas}(k)\overline{n}_{I,i}^{tot}\left(1+\epsilon_{I,i}(k)\right) \tag{12}$$

$$c_i(k) = \overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)\mathbb{I}_{\left\{b_i(k) \ge \overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)\right\}} h_{seas}(k)\overline{n}_{I,i}^{tot}\left(1+\epsilon_{I,i}(k)\right). \tag{13}$$

For future reference, pay attention to how $n_{I,i}(k)$ and $c_i(k)$ depend on $b_i(k)$ for different values of all other variables. For each time k and specific realization of all other variables, $n_{I,i}(k)$ and $c_i(k)$ are single step functions of $b_i(k)$. However, the error variables $\epsilon_{b^\star,i}(k)$ and $\epsilon_{I,i}(k)$ impact the above relationships in fundamentally different ways. The error term $\epsilon_{b^\star,i}(k)$ introduces a horizontal shift of both $n_{I,i}(k)$ and $c_i(k)$ as step functions of $b_i(k)$, and introduces a vertical scaling of $c_i(k)$. Meanwhile, $\epsilon_{I,i}(k)$ scales both $n_{I,i}(k)$ and $c_i(k)$ vertically. We return to this important fact later.

Value Generation describes the relationship from awarded impression volume $n_{I,i}(k)$ to sourced value $v_i(k)$. We assume that each impression independently leads to an event with probability p_i. In other words, the number of events $n_{E,i}(k)$ is a realization of a random variable $N_{E,i}(k)$, which is distributed as Binomial$(n_{I,i}(k), p_i)$ with an expected value of $n_{I,i}(k)p_i$ and variance $n_{I,i}(k)p_i(1-p_i)$, which under the assumption $p_i \ll 1$ approximately equals $n_{I,i}(k)p_i$. Actually, in some applications where each impression may turn into more than one event, a more appropriate model may be $N_{E,i}(k) \sim$ Poisson$(n_{I,i}(k)p_i)$. However, for sufficiently large values of $n_{I,i}(k)$ and small values of $n_{I,i}(k)p_i$ we know from the Central Limit Theorem that the binomial and Poisson models are virtually identical, so in practice it makes no difference which model we use. To describe the value generation subsystem in a convenient input–output form, we propose an event volume model $n_{E,i}(k) = p_i n_{I,i}(k)(1+\epsilon_{E,i}(k))$, where $\epsilon_{E,i}(k)$ is an independent random variable with mean zero and variance $\sigma_{E,i}^2 = 1/\sqrt{p_i n_{I,i}(k)}$. The value added to the campaign from segment i is given by the number of events times the monetary value per event v_E. Putting things together yields

$$v_i(k) = v_E p_i(1+\epsilon_{E,i}(k))n_{I,i}(k). \tag{14}$$

Conversion Latency captures the time delay from a served ad impression to a measured event. It models the fact that Internet users often do not respond to a seen ad immediately. For impressions and clicks, the latency is almost exactly the data flow delay from ad serving system to control system. This delay is often somewhere between 15 min and an hour. For conversions, on the other hand, the latency may be in the order of hours or days. Fig. 4

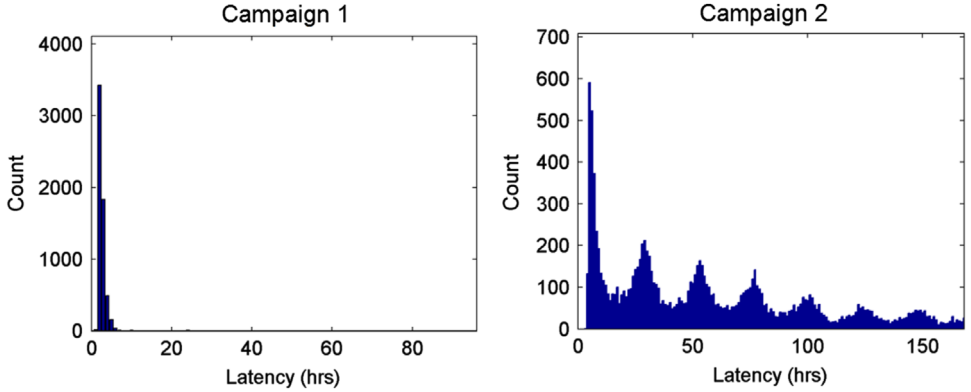

Fig. 4. Examples of experimental latency data. Campaign 1 is characterized by low-latency events, while Campaign 2 is characterized by long-latency events.

provides representative examples of the latency distribution for two different ad campaigns. Each plot is a histogram of the number of events with a certain latency.

For Campaign 1 we see that all events have a latency of at most a few hours, while for Campaign 2 there are many events with a latency of 100–150 h, or more. For Campaign 2 we also see a decaying 24 h-periodic pattern, which is an artifact of Internet user behavior. For example, a user may have seen an ad at 8am on day 1. If the user does not respond to the ad right away, it is more likely (if at all) that the user responds after 24 h rather than sooner. The latency distribution from one campaign to another may differ a lot, but tends to be approximately the same across segments within one campaign. Regardless, an accurate model of the latency is quite complex and a priori unknown. Meanwhile, since the model must be estimated online with a poor initial estimate, we recommend a very simple model structure, e.g., a first order linear and time-invariant model that can be expressed by

$$y_{value,i}(k) = h_{latency}(q)v_i(k), \tag{15}$$

where $h_{latency}(q)$ is a first order linear function of the shift operator q defined by $qv_i(k) = v_i(k+1)$.

Cost Latency describes the relationship between actual cost $c_i(k)$ and measured cost $y_{cost,i}(k)$, and is dictated by the time it takes to identify highest competing bids, sum them up, and feed the summed value to the control system. This delay is relatively short and is modeled by a simple one-step time delay; i.e.,

$$y_{cost,i}(k) = c_i(k-1). \tag{16}$$

Having modeled each subsystem separately, we now consider their interconnection.

A *Campaign Model* is obtained by combining the above subsystem models. Simply adjoin the subsystem input–output relationships defined by (11)–(16) and sum the segment-level outputs over all segments:

$$n_I(k) = h_{seas}(k)h_{imps}\left(u, p_i, \overline{b}_i^\star, \epsilon_{\hat{p},i}, \epsilon_{b^\star,i}, \epsilon_{I,i}, \forall i\right)$$
$$v(k) = h_{seas}(k)h_{value}\left(u, p_i, \overline{b}_i^\star, \epsilon_{\hat{p},i}, \epsilon_{b^\star,i}, \epsilon_{I,i}, \epsilon_{E,i}, \forall i\right)$$
$$c(k) = h_{seas}(k)h_{cost}\left(u, p_i, \overline{b}_i^\star, \epsilon_{\hat{p},i}, \epsilon_{b^\star,i}, \epsilon_{I,i}, \forall i\right)$$

where (under the assumption $p_i(1+\epsilon_{\hat{p},i}(k)) > 0$)

$$h_{imps}(\cdot) = \sum_i \mathbb{I}_{\left\{u(k) \geq \frac{\overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)}{p_i\left(1+\epsilon_{\hat{p},i}(k)\right)}\right\}} \overline{n}_{I,i}^{tot}\left(1+\epsilon_{I,i}(k)\right) \tag{17}$$

$$h_{value}(\cdot) = v_E \sum_i p_i\left(1+\epsilon_{E,i}(k)\right)\mathbb{I}_{\left\{u(k) \geq \frac{\overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)}{p_i\left(1+\epsilon_{\hat{p},i}(k)\right)}\right\}} \overline{n}_{I,i}^{tot}\left(1+\epsilon_{I,i}(k)\right) \tag{18}$$

$$h_{cost}(\cdot) = \sum_i \overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)\mathbb{I}_{\left\{u(k) \geq \frac{\overline{b}_i^\star\left(1+\epsilon_{b^\star,i}(k)\right)}{p_i\left(1+\epsilon_{\hat{p},i}(k)\right)}\right\}} \overline{n}_{I,i}^{tot}\left(1+\epsilon_{I,i}(k)\right) \tag{19}$$

The measured outputs are obtained by applying the latency dynamics $y_{value}(k) = h_{latency}(q)v(k)$ and $y_{cost}(k) = c(k-1)$.

The expressions for $h_{imps}(\cdot)$, $h_{value}(\cdot)$, and $h_{cost}(\cdot)$ are intimidating at first, but provide a wealth of insights helping us to understand the optimization and control problem. For example, the model demonstrates how the statistical properties of error signals $\epsilon_{\hat{p},i}(k)$, $\epsilon_{b^\star,i}(k)$, $\epsilon_{E,i}(k)$ translate into different statistical properties of $v(k)$, $c(k)$, $n_I(k)$, and $r(k)$ $(= v(k)/c(k))$.

With known empirical or estimated distributions of all campaign model parameters (latency and seasonality parameters, highest competing bid prices, error distributions, etc.), the plant model is immediately useful during the design of a feedback controller: Given the above distributions, we can estimate bounds on the effective steepness (or "plant gain") of $h_{value}(\cdot)$ and $h_{cost}(\cdot)$ as functions of u, and bounds on the parameters describing the latency and seasonality distributions. The control problem can then be expressed in the framework of robust control, where performance, stability, and robustness is traded against each other.

On the other hand, since the properties of $\epsilon_{\hat{p},i}(k)$ are heavily dependent on the event rate estimation algorithm we design, we can influence the plant behavior in a positive way by selecting an estimation algorithm with good properties. Good properties lead to a well-behaved interconnected system that solves the overarching optimization problem with high performance ($b_i(k) = u(k)\hat{p}_i(k) \approx u^{opt}p$), stability, and desired level of robustness.

Note, there are many important properties of estimation algorithms to consider beyond e.g. bias and variance. For example, suppose estimation algorithms A and B result in error processes that can be modeled by $\epsilon_{\hat{p},i}^A(k) = \omega(k)$ and $\epsilon_{\hat{p},i}^B(k) = 0.8\epsilon_{\hat{p},i}^B(k-1) + 0.6\omega(k)$, respectively, where $\omega(k)$ is independent and identically distributed with mean zero and variance one. It is easy to show that $\mathrm{E}\left(\epsilon_{\hat{p},i}^A(k)\right) = \mathrm{E}\left(\epsilon_{\hat{p},i}^B(k)\right) = 0$ and $\mathrm{Var}\left(\epsilon_{\hat{p},i}^A(k)\right) = \mathrm{Var}\left(\epsilon_{\hat{p},i}^B(k)\right) = 1$, while $\mathrm{Cov}\left(\epsilon_{\hat{p},i}^A(k), \epsilon_{\hat{p},i}^A(k-1)\right) = 0$ and $\mathrm{Cov}\left(\epsilon_{\hat{p},i}^B(k), \epsilon_{\hat{p},i}^B(k-1)\right) = 0.48$. That is, event rate estimates $\hat{p}_i^A(k)$ and $\hat{p}_i^B(k)$ are both unbiased and have identical variance, but they have very different auto-covariance which may suggest one of the estimators is superior for an integrated system. There may be other similarities and differences between the estimators that should also be considered, for example related to higher order statistical moments, but the key message is that the system-level impact must be considered when designing subcomponents for an online advertising optimization system.

Furthermore, to the extent competing campaigns are also managed by us, it is appropriate to use insights obtained from the model to come up with bidding strategies that promote a healthy interaction across campaigns and a sound overall network behavior, and not only good behavior for our first campaign. Under all circumstances, the campaigns and segments in reality do not operate in a vacuum, and there is plenty of coupling across the network. For example, to a lesser or larger degree, the ad delivery in segment i impacts the delivery in a segment j since our bidder changing $b_i(k)$ may trigger a competing bidder to change their bids in multiple segments causing a change of $b_j^\star(k)$ (the highest competing bid in a different segment), etc.

To further enhance our appreciation of the plant behavior and contrast some results in Section 2.1, it can be shown that with the bidding strategy $b_i = u\hat{p}_i$ (rather than $b_i = up_i$), $c = c(u)$ and $v = v(u)$ remain monotonic (non-decreasing) functions while $r = r(u)$ potentially is non-monotonic. It is trivial to prove monotonicity of $h_{value} = h_{value}(u)$ and $h_{cost} = h_{cost}(u)$ in the presence of event rate estimation error. However, with non-zero event rate estimation error $\epsilon_{\hat{p},i}$ we can easily construct examples where $r(u)$ is decreasing in some intervals of u, and increasing in others. Indeed, the monotonicity proof for $r(u)$ in Appendix B relies on $b_i = up_i$.

Clearly, $h_{imps}(u)$, $h_{value}(u)$, and $h_{cost}(u)$ (and therefore also $n_I(u)$, $v(u)$, $c(u)$, and $r(u)$) are, strictly speaking, staircase functions. The significance of this fact and whether we can ignore the discontinuities depends on the density and height of each step specified by (17)–(19).

To proceed with this assessment, we can either use a *bottom-up* or a *top-down* approach. In the bottom-up approach we analyze empirical distributions of all campaign model parameters, such as highest competing bid prices, error distributions, etc. (as

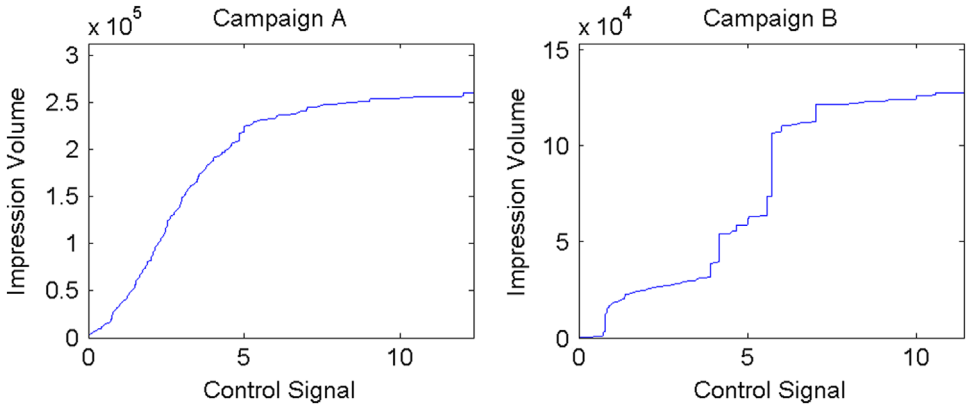

Fig. 5. Two representative examples of impression volume $h_{imps}(u)$ versus control signal u showing what is the expected number of impressions for different values of the control signal u. Campaign A is a well-behaved campaign with an approximately smooth relationship while Campaign B is a challenging campaign with pronounced large steps in the relationship.

suggested earlier) to reconstruct or simulate plausible stair case functions. On the other hand, in the top-down approach we consider a short time interval and inspect all the price points at which impressions were sold for different segments. From there we reconstruct the stair case functions.

It is outside the scope of this paper to dig deep into this matter, and we simply produce two sample staircase functions according to the top-down approach based on empirical data. Fig. 5 shows two representative examples of $h_{imps}(u)$, where each plot is produced based on one day's worth of impression data. As a side note, since gathering and processing the low-level impression data is computationally very expensive, it would not be feasible to produce the plots in run-time and at scale to a control system. Producing and analyzing the plots is primarily useful for off-line modeling.

Campaign A is nonlinear and discontinuous, but each individual step in $h_{imps}(u)$ is so small that the function in practice can be treated as smooth. The slope of the curve (the plant gain), however, varies over a wide range. At a control signal of \$3, the slope is about 60,000 impressions/\$, while at a control signal of \$12, the slope is close to zero.

Campaign B, on the other hand, has some very distinct large isolated steps as a result of dominant segments. This means that for a campaign delivery goal of, say, 90,000 impressions, there exists no fixed control signal maintaining the delivery goal. The only way to hit the goal, on average, is to alternate between under- and over-pacing. This creates a volatile delivery of the campaign and is likely to result in limit cycles or chaos of the closed-loop system. Furthermore, a volatile delivery of Campaign B may also create a ripple effect across the network since the volatility in our bid prices and the impression allocation appears as volatile competing bid prices and impression allocation to other campaigns. It is natural to ask what the likelihood is to operate on a step. Note, Campaigns A and B are both representative examples of advertising campaigns. With this in mind, assume to a first order approximation, that any delivery goal between 0 and 130,000 for Campaign B is equally likely. Since the large step stretch from approximately 60,000 to 110,000, there is a probability of about $5/13 = 0.38$ that the optimal operating point is on the step. In other words, operating on a step far from a plateau is a quite likely scenario.

Another important aspect of the steps in the plant relate to how they move over time. Error term $\left(1+\epsilon_{b^\star,i}(k)\right)/(1+\epsilon_{\hat{p},i}(k))$ in (17) shifts step i in the $h_{imps}(u)$ plot horizontally, which for one of the large steps could have a dramatic impact on the campaign delivery, and in the extension, on the network behavior due to ripple effects. Error term $\epsilon_{l,i}(k)$ scales step i in the plot vertically, which in general leads to a less dramatic effect.

In addition to the shape, shifts, and scaling of the staircase function, we must keep the feedback latency in mind. The latency of $c(k)$ and $n_l(k)$ is approximately zero, but if the value-bearing event is a conversion the latency of $v(k)$ may be in the order of many hours or days, where the latency distribution is a priori unknown and must be estimated online. Finally, since there is no measured equivalence of $r(k)$, it must be estimated by the controller from $y_{value}(k)$ and $y_{cost}(k)$ while accounting for the latency encoded in $y_{value}(k)$.

2.3. Challenges

The plant model developed in the previous section can be used to identify challenges in solving the optimization problem defined in Section 2.1. Some of these challenges are outlined in this section, but while discussing these challenges keep in mind that for scalability reasons we are only interested in a decoupled solution where segment-level event rate estimation and campaign-level feedback control are handled separately. Moreover, optimality requires that each $\hat{p}_i$ and u converge to, and remain near, p_i and u^{opt}, respectively.

Consider first the segment-level event rate estimation. Computing $\hat{p}_i(k)$ is difficult for several reasons: The unknown p_i is usually very small ($\ll 1$), which means a large number of impressions is needed to observe an adequate number of events for accurate event rate estimation. Meanwhile, an estimate $\hat{p}_i(k)$ may lead to a bid price $b_i(k) = u(k)\hat{p}_i(k)$ that is not competitive enough to win any impression at all, which precludes enhanced estimation. Therefore, to win impressions, the event rate estimate is often artificially inflated before the bid price is computed; however, the highest competitive bid price is unknown and the system delay is at least one sample time. The end-result at time k may be no awarded impression and no enhanced estimate, or alternatively, a very large number of awarded impressions with a detrimental effect on the campaign-level ROI. An additional challenge springs from the unknown event latency. A small number of generated events early on in a campaign may be the result of either a small event rate or a long latency, so event rate and latency distribution must be estimated simultaneously and the estimation is an ill-conditioned regression problem until a sufficiently long time-series of impression and event data is available.

Campaign-level feedback control is difficult for a number of reasons, e.g. related to the imperfect segment-level event rate estimates, the discontinuous input–output relationship of the plant, the time-varying property of the plant, the unknown and

vastly different plant gains and latency distributions, and finally the dynamic coupling of bidders.

Poor or artificially inflated segment-level event rate estimates, which are particularly common in the early phase of a campaign, may lead to a non-monotone $r(u)$ triggering a destabilizing positive feedback in the campaign-control loop. Furthermore, because of the very small true event rates p_i, it is commonplace that the relative event rate estimation error $\epsilon_{\hat{p},i}$, defined by $\hat{p}_i = p_i(1-\epsilon_{\hat{p},i})$, has a standard deviation of, say, 0.1 or more even days or weeks into a campaign. This large relative error leads to significant horizontal shifting of the steps in the functions defined by $h_{imps}(u)$, $h_{value}(u)$, and $h_{cost}(u)$, and frequently causes dramatic and alternating delivery spikes and dips in the campaign-level feedback signals.

The auction-based allocation of impressions implies that $h_{imps}(u)$, $h_{value}(u)$, and $h_{cost}(u)$ describe staircase functions, which translates into input–output relationships of the plant that are discontinuous. This makes the vast majority of results in the control theory literature non-applicable. Moreover, the distribution of $b_i^\star/\hat{p}_i$'s and $n_{l,i}$'s dictates the density and height of the steps in the staircase functions and may vary a lot from one campaign to the next. A staircase for some campaigns, and in some intervals of u, inclines rapidly. For other campaigns, or in other price intervals for the same campaign, the incline is virtually zero. So, even in the best case scenario, where the plant input–output relationship is approximately smooth thanks to a large number of small steps, the effective and a priori unknown plant gain (defined by the rate of incline) may be very small or very large.

It is well-known that the stability of a closed-loop system is tightly coupled to the relationship between plant and controller gains due to the Nyquist stability criterion. Hence, since the plant gain for a campaign varies across campaigns and price points, and over time, it is necessary to estimate the effective plant gain adaptively. However, the direct or indirect estimation of a plant gain is a difficult problem since it requires persistent excitation [3]. It is particularly difficult when the underlying true plant is discontinuous. We deal with an extreme plant where the slope, strictly speaking, is either zero or infinity, which compounded with the various noise described earlier makes the control and estimation problem especially challenging.

The common scenario where no fixed point solution exists is related to the staircase plant. This scenario is present when the desired delivery is between two steps. As pointed out in Section 2.1, the optimal solution is to make use of fractional bidding, but the provided formula (10) for computing the optimal fraction (or, bid allocation) requires segment level information to a degree and accuracy that is not practically possible.

One more perspective of the discontinuous plant relates to time-varying competitive bids. The highest competitive bid may change gracefully or abruptly as a result of other bidders entering or leaving the competition, or simply because of fluctuations in their event rate estimates or control signals.

The time-varying (approximately periodic) pattern in Internet traffic, and sometimes in competing bids, makes optimal bidding and smooth delivery of the campaign budget difficult. Since a too fast or a too slow delivery of a campaign budget near the traffic peak one day is difficult to compensate for until the following day because of many hours of very low traffic.

Finally, the unknown event latency distribution contributes to the complexity of campaign-level control in a manner similar to segment-level event rate estimation. Since a small number of generated events early on in a campaign may be the result of either a small actual event rate or a long latency, the estimated total ROI r must account for the latency. Indeed, ROI r and latency distribution $h_{latency}(q)$ must be estimated simultaneously based on y_{value} and y_{cost} and is an ill-conditioned estimation problem.

3. Randomized bidding

3.1. Gamma distribution

The gamma distribution from mathematical statistics is a continuous probability distribution with parameters α and β. If the random variable B follows the gamma distribution, then we write $B \sim \text{Gamma}(\alpha,\beta)$. The probability density function of b is given by

$$f(b|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} b^{\alpha-1} e^{-\beta b}$$

for $b \geq 0$, where $\Gamma(\alpha)$ is the gamma function defined by $\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1} dt$. Parameters $\alpha > 0$ and $\beta > 0$ are referred to as *shape* and *inverse scale*. The expected value of B is $\text{E}(B) = \alpha/\beta$, while the variance is $\text{Var}(B) = \alpha/\beta^2$.

Fig. 6 shows three examples of gamma probability density functions. Note that the density functions corresponding to the black and red curves have the same expected values, while the black curve has a larger variance.

The probability that B is larger than or equal to $b^\star$ is obtained by integrating $f(b|\alpha,\beta)$ from $b^\star$ to infinity; i.e.,

$$\text{Prob}(B \geq b^\star) = \int_{b^\star}^\infty \frac{\beta^\alpha}{\Gamma(\alpha)} B^{\alpha-1} e^{-\beta B} dB. \tag{20}$$

Moreover, if $B \sim \text{Gamma}(\alpha,\beta)$, then $B\beta \sim \text{Gamma}(\alpha,1)$ [9]. Hence,

$$\begin{aligned}\text{Prob}(B \geq b^\star) &= \text{Prob}(B\beta \geq b^\star\beta) \\ &= \text{Prob}\left(\tilde{B} \geq b^\star\beta\right),\end{aligned} \tag{21}$$

where $\tilde{B} \sim \text{Gamma}(\alpha,1)$, which is used extensively in Section 4.

3.2. Heisenberg bidding

The idea of using bid randomization for improved control and estimation in applications involving a large number of auctions was proposed in [17]. It was coined *Heisenberg bidding* as a catch phrase to illustrate how its impact on bidding superficially resembles the tunneling effect in quantum mechanics. The analogue in auctions is that an otherwise non-competitive bid may outbid a nominally higher bid.

Heisenberg bidding operates by randomly perturbing a *nominal bid price* b_p according to a distribution defined by b_p and a *bid uncertainty* b_u, to generate a *final bid price* b used in the market clearing, where b is a realization of a random variable B. Heisenberg bidding can be implemented with other probability

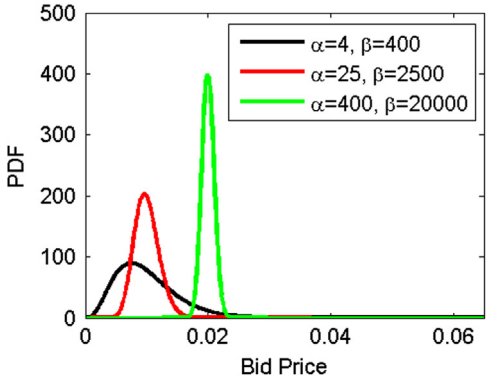

Fig. 6. Three examples of gamma probability density functions parameterized by the shape and inverse scale parameters. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

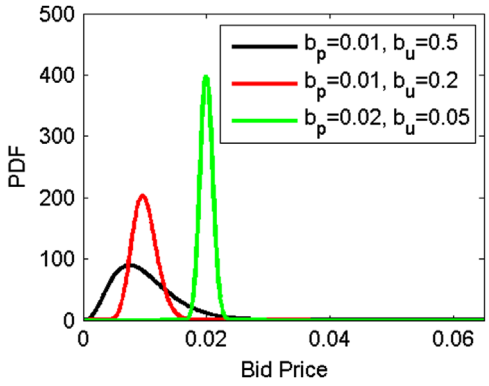

Fig. 7. Three examples of gamma probability density functions parameterized by the nominal bid price b_p and the bid uncertainty b_u.

distributions, but in this paper it is defined by

$$B \sim \text{Gamma}\left(\frac{1}{b_u^2}, \frac{1}{b_p b_u^2}\right) \quad \text{if } b_p, b_u > 0 \tag{22}$$

and $B = b_p$, otherwise. Note, in terms of the shape parameter α and the inverse scale parameter β, Heisenberg bidding with the gamma distribution is given by $\alpha := 1/b_u^2$ and $\beta := 1/(b_p b_u^2)$. It follows that $\text{E}(B) = b_p$ and $\text{Var}(B) = b_p^2 b_u^2$. With the proposed parameterization b_u defines the unit-less relative standard deviation of B; i.e., $\text{Std}(B)/\text{E}(B)$, where $\text{Std}(B) = \sqrt{\text{Var}(B)}$.

Fig. 7 shows the same three examples of gamma distributions as in Fig. 6, but here the distributions are parameterized by b_p and b_u. The bid uncertainty b_u defines the level of perturbation of the nominal bid. If $b_u = 0$ there is no perturbation at all.

3.3. Sample use cases

Heisenberg bidding is versatile and can be used in many ways. In Section 4 we demonstrate how it is valuable for efficient exploration and exploitation (including event rate estimation). The algorithm provides ongoing exploration of low-performing bidders while reliably exploiting high-performers. Moreover, it is adaptive and robust to gradual dynamic changes in the competitive landscape, and less prone to undesired impression volume spikes than conventional techniques, since it never artificially inflates all bids to a high level that may exceed a highest competing bid for a large number of impressions.

Another use case of Heisenberg bidding is to implement the fractional bidding in (10) without making use of sub-campaign information. This is accomplished by computing a nominal control signal $u_p(k)$ using feedback control like before, but to supplement it with a small control signal uncertainty $u_u(k)$. The final control signal u is generated for each impression request as a realization of the random variable U, where $U \sim \text{Gamma}(1/u_u^2, 1/(u_p u_u^2))$, and used to calculate the bid price according to $b_i = u\hat{p}_i$. Since u is a randomly generated value, b_i is sometimes larger and sometimes smaller than $b_i^\star$, effectively resulting in fractional bidding. While the perturbation applies to all segments i, as long as u_u is sufficiently small and $u_p \approx u^{opt}$, the fractional bidding is significant only in the segment where $b_i \approx b_i^\star$, which happens to be the segment among those where we are competitive with the lowest ROI impressions. This is the segment, which in Section 2.1 was denoted i'. Hence, with good choices of u_p and u_u we obtain

$$a_i = \text{Prob}(U\hat{p}_i \geq b_i^\star) \approx \begin{cases} 1, & \text{if } i = 1, \ldots, i'-1 \\ 0, & \text{if } i = i'+1, i'+2, \ldots \end{cases}$$

and where $a_{i'}$ is approximately the bid allocation value given for segment i' in (10).

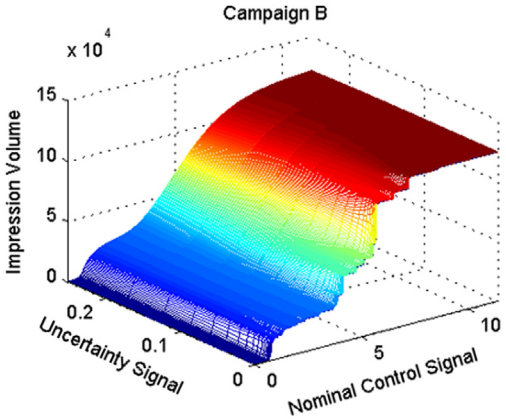

Fig. 8. The impact of using bid uncertainty on Campaign B in Fig. 5. Note that the nominal control signal versus impression volume relationship is discontinuous only when the uncertainty signal equals zero.

Heisenberg bidding may also be used more generally to smoothen the otherwise discontinuous input–output relationships of the plant and remove the extreme plant gains (turn infinite plant gains finite, and zero plant gains non-zero), thereby improving the conditions for feedback control. This is illustrated in Fig. 8, which shows the result of adding the dimension of uncertainty signal u_u to Campaign B in Fig. 5. Note, how the input–output relationship of the plant can be made arbitrarily smooth and effectively linear with a sufficiently large value of u_u. From a control design perspective, the use case of increasing and decreasing the plant gain is a valuable complement to the more common technique in adaptive control of adjusting only the controller gain. Ultimately, the closed loop performance and stability depends on the loop gain, which is the product of controller and plant gains. However, keep in mind there is a trade-off involved since the improved condition for feedback control also means an increasing number of bids are submitted far from the optimal up_i, resulting in undesirable fractional bidding in segments other than i'.

With some amount of Heisenberg bidding applied to a bidder's nominal bid, the bidder detects a competing bid that is approaching b_i from below or above gradually rather than abruptly. This helps to prevent sudden impression spikes and dips and means the bid randomization may improve the performance and robustness of a time-varying network.

All in all, Heisenberg bidding facilitates a harmonic interaction among bidders and makes it easier to deal with many of the challenges described in Section 2.3.

4. Exploration and exploitation

4.1. Problem formulation

The objective is to estimate an event rate p in a network of bidding agents in a manner that ensures ongoing low cost reinforced learning, facilitates a harmonic interaction with other bidders, and gracefully handles system delays and dynamic competitive landscapes.

Reinforced learning in the context of auction-based networks is often described in terms of so-called exploration and exploitation: The bidding strategy should, on one hand, properly reflect the best current estimate of the event rate to maximize the expected short term value (exploitation). On the other hand, it should support ongoing learning (exploration) to maximize the expected long term value. The exploration requires some relatively high bids for impressions that appear to have low ROI. In particular, in order to win impressions and observe events necessary to improve the

event rate estimate, some bids may need to be higher than $b = u^{opt}\hat{p}$, where $\hat{p}$ is our best current estimate of p.

Trading exploration and exploitation is a challenging problem. The vast majority of solutions proposed over the years make unrealistic assumptions such as zero system delay, constant p, static competitive landscape, infinite time horizon, and no objective beyond exploration and exploitation. Many or all of these assumptions are violated in real problems. Indeed, in an online advertising system the actuator and measurement delays commonly sum to at least a few minutes but sometimes to an hour or more. Furthermore, the underlying true event rate, as well as the competing bids, normally change over time. Also, all real applications have a finite time horizon, and very often the event rate estimator system is integrated with, e.g., a control system, introducing a dynamic interaction between the subsystems that cannot be disregarded.

In this section we propose a solution to one part of the optimization problem defined in Section 2.1. In particular, we develop an algorithm to compute $\hat{p}$ (where we have dropped subscript i to avoid unnecessary clutter). The algorithm is designed to address some of the challenges discussed in Section 2.3, e.g., enable a harmonic interaction with other bidders and handle modest system delays and dynamic competitive landscapes. Our solution makes use of randomized bidding (Section 3) and demonstrates one use case of Heisenberg bidding.

4.2. Bayesian event rate estimation

We first derive an optimal Bayesian event rate estimator to show that the exploration and exploitation algorithm presented later is more than just a convenient choice and to tie together the principle of Heisenberg bidding with the proposed algorithm for exploration and exploitation.

Suppose the event volume $n_E(k)$ is a realization of the random variable $N_E(k) \sim \text{Binomial}(n_I(k), p)$, which under the assumption $n_I(k) \geq 100$ and $n_I(k)p \leq 10$ by virtue of the Central Limit Theorem [9] can be well approximated using a Poisson distribution

$$N_E(k) \sim \text{Poisson}\,(n_I(k)p). \tag{23}$$

Consider one pair of observations n_I and n_E and leave out the time index k until later. The probability mass function of n_E based on the Poisson model is given by

$$f(n_E \mid n_I, p) = \frac{(n_I p)^{n_E} e^{-n_I p}}{n_E!}.$$

Given a prior belief function $\pi(p)$ for p, we proceed by using Bayes rule to determine a posterior $\pi(p \mid n_I, n_E)$ that incorporates the evidence of p encoded in n_I and n_E.

Since the conjugate prior of the Poisson probability mass function is given by the gamma probability density function, we assume that the prior belief function $\pi(p)$ is given by $\text{Gamma}(\alpha_0, \beta_0)$. A standard application of Bayes rule $\pi(p \mid n_I, n_E) = f(n_E \mid n_I, p) f(p \mid n_I) / f(n_E \mid n_I)$ shows that $p \mid n_I, n_E \sim \text{Gamma}(\alpha_0 + n_E, \beta_0 + n_I)$. This update scheme applies recursively for each new set of measurements $n_I(k), n_E(k)$, $k = 1, 2, \ldots$. Therefore,

$$p \mid n_I(1:k), n_E(1:k) \sim \text{Gamma}\left(\alpha_0 + \sum_{i=1}^{k} n_E(i), \beta_0 + \sum_{i=1}^{k} n_I(i)\right),$$

or equivalently by defining the shape and inverse scale parameters as a time-invariant Markov process with state $[\alpha(k), \beta(k)]^T$

$$\begin{bmatrix} \alpha(k) \\ \beta(k) \end{bmatrix} = \begin{bmatrix} \alpha(k-1) \\ \beta(k-1) \end{bmatrix} + \begin{bmatrix} n_E(k) \\ n_I(k) \end{bmatrix},$$

and initialized by $[\alpha(0), \beta(0)]^T = [\alpha_0, \beta_0]^T$, the posterior distribution of p at time k is given by $\text{Gamma}(\alpha(k), \beta(k))$.

However, if the unknown p might be slowly varying, it is necessary to discount old observations and rely more on recent observations in the estimator. We do this by introducing a memory loss specified by a forgetting factor λ, where $0 < \lambda < 1$. In particular, we adopt the following revised update scheme according to $\alpha(k) = \lambda^k \alpha_0 + \sum_{i=1}^{k} \lambda^{k-i} n_E(i)$ and $\beta(k) = \lambda^k \beta_0 + \sum_{i=1}^{k} \lambda^{k-i} n_I(i)$. This leads to the posterior distribution

$$p \mid n_I(1:k), n_E(1:k) \sim \text{Gamma}\left(\lambda^k \alpha_0 + \sum_{i=1}^{k} \lambda^{k-i} n_E(i), \lambda^k \beta_0 + \sum_{i=1}^{k} \lambda^{k-i} n_I(i)\right),$$

or, in recursive form:

$$\alpha(k) = \lambda \alpha(k-1) + n_E(k) \tag{24}$$

$$\beta(k) = \lambda \beta(k-1) + n_I(k) \tag{25}$$

The posterior expected loss given a squared loss function of $L(p, \hat{p}) = (p - \hat{p}(k))^2$ is defined by

$$\mathrm{E}^{\pi(p \mid n_I(1:k), n_E(1:k))}(L(p, \hat{p})) = \int_0^\infty (p - \hat{p}(k))^2 \pi(p \mid n_I(1:k), n_E(1:k)) dp$$

Minimizing $\mathrm{E}^{\pi(p \mid n_I(1:k), n_E(1:k))}(L(p, \hat{p}))$ with respect to $\hat{p}(k)$ yields

$$\hat{p}(k) = \frac{\alpha(k)}{\beta(k)}. \tag{26}$$

This incorporates all available evidence into the estimate and defines the optimal Bayesian event rate estimate. The posterior variance $\hat{\sigma}^2$ of p at time k is defined by $\hat{\sigma}^2(k) = \mathrm{Var}^{\pi(p \mid n_I(1:k), n_E(1:k))}(p)$, which for the gamma posterior distribution implies that

$$\hat{\sigma}^2(k) = \frac{\alpha(k)}{\beta^2(k)}.$$

It is insightful to plug the closed form expressions for $\alpha(k)$ and $\beta(k)$ into the estimators $\hat{p}(k)$ and $\hat{\sigma}^2(k)$ and explore their dependency of the observations. In particular, suppose for a moment that no impression or event data is recorded after time k', where $k' < k$. Then $\alpha(k) = \lambda^{k-k'} \alpha(k')$ and $\beta(k) = \lambda^{k-k'} \beta(k')$, which leads to

$$\hat{p}(k) = \hat{p}(k')$$
$$\hat{\sigma}^2(k) = \lambda^{k'-k} \hat{\sigma}^2(k') > \hat{\sigma}^2(k').$$

In other words, the event rate point estimate $\hat{p}(k)$ remains unchanged, while the variance $\hat{\sigma}^2(k)$ gradually increases whenever no impression or event data is recorded. This is an intuitive result, but also a key property utilized in the exploration and exploitation algorithm presented next.

In conclusion, the event rate estimator is defined by the second order, nonlinear, and time-invariant difference equation given by (24)–(26).

4.3. Principle idea

The basic idea of the proposed exploration and exploitation algorithm is to implement Bayesian event rate estimator (24)–(26) using the sample rate of the impression and event volume data flow (e.g. with new data every 5 or 15 min), and to use the Heisenberg bidding mechanism (22) for each individual impression request (aka auction) to generate a bid price from the most up-to-date posterior distribution of the event rate. Because the total number of impression requests per hour and segment in online advertising tends to be large and in the order of thousands or millions, the end result is practically a continuum of bids with the highest density of bids at or near the Bayesian expected event rate.

The state variables $\alpha(k)$ and $\beta(k)$ of the Bayesian estimator separately give no indication of what the true event rate is and provide little value to an operations engineer monitoring the system. For that reason and to simplify the integration with a potential control adjustment $u(k)$, we recommend the output of

the estimator to be the posterior expected value and relative standard deviation $[b_p(k), b_u(k)]$, which are determined uniquely and one-to-one from $[\alpha(k), \beta(k)]$. Note, in Section 3.2 and (22), we refer to $[b_p(k), b_u(k)]$ as the nominal bid price and bid uncertainty. In other words, the output of the estimator is

$$b_p(k) = \alpha(k)/\beta(k) \tag{27}$$

$$b_u(k) = 1/\sqrt{\alpha(k)} \tag{28}$$

where $\alpha(k) \neq 0$ and $\beta(k) \neq 0$. In an integrated system consisting of a segment-level estimator (or, exploration and exploitation algorithm) and a campaign-level feedback controller, the output from the estimator typically is $\hat{p}_i(k) = \alpha_i(k)/\beta_i(k)$ and $\hat{\sigma}_i(k) = 1/\sqrt{\alpha_i(k)}$, while the output from the feedback controller is $u_p(k)$ and $u_u(k)$. The nominal bid price and bid uncertainty used for the bid randomization can then be defined by $b_p(k) = u_p(k)\hat{p}_i(k)$ and $b_u(k) = \hat{\sigma}_i(k) + u_u(k)$.

However, for the remainder of this paper we only discuss exploration and exploitation and assume $u_p(k) = 1$ and $u_u(k) = 0$. To save space we therefore do not explicitly spell out the mapping (27)–(28), but remind the reader that the mapping is used in any practical implementation for the reasons explained above.

To illustrate the basic principle of Heisenberg bidding-based exploration and exploitation, consider two bidders A and B that are interested in the same impressions. Each one is submitting bids according to a bid distribution updated using the above Bayesian update scheme, where each impression is awarded to the highest final bid. Denote A's event rate p_A and B's event rate p_B, and suppose the value of each event of interest (e.g. impression, click, or conversion) for both bidders equals one.

Fig. 9 illustrates how the bid distribution of A evolves if p_A is larger than p_B. The red area shows the approximate probability of bidder A to win an auction. The Bayesian estimator ensures that the average bid price converges to p_A ($> p_B$). The large value of p_A results in the bidder being awarded the majority of impressions, implying that the confidence in the estimated value for p_A grows and the posterior distribution of p_A becomes increasingly spiky.

Fig. 10 illustrates how the bid distribution of A evolves if A is lower performing than B. The Bayesian estimator ensures that the average bid price converges to p_A ($< p_B$). The relatively low value of p_A may temporarily result in no impressions being awarded, but the variance-increasing property of the estimator ensures that an equilibrium is eventually reached, where a few impressions are awarded to A.

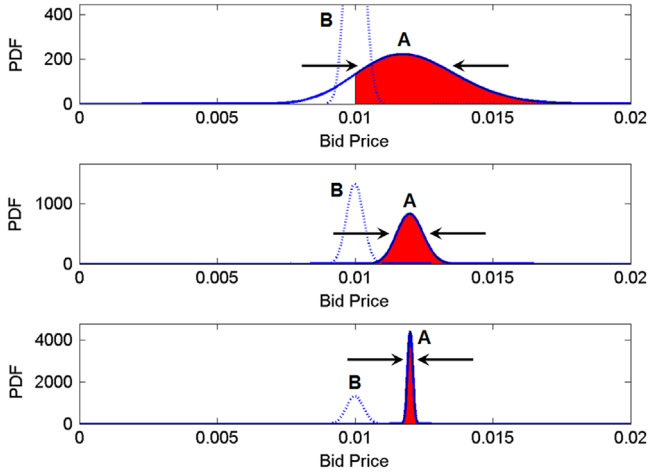

Fig. 9. Qualitative illustration of how the bid distribution evolves for a high-performing bidder. The distribution becomes increasingly spiky. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

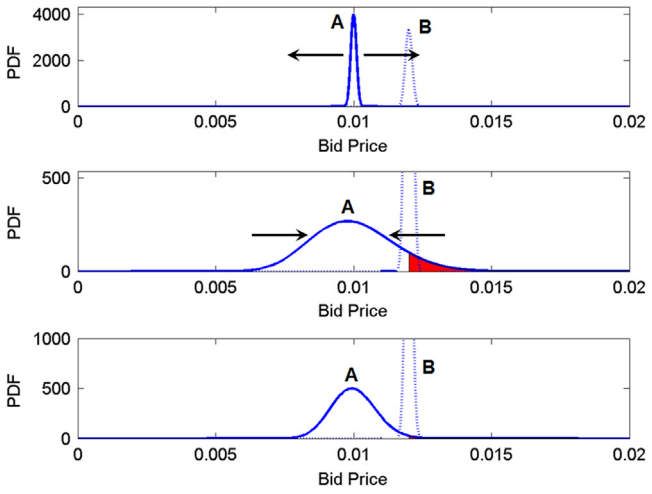

Fig. 10. Qualitative illustration of how the bid distribution evolves for a low-performing bidder. The distribution eventually reaches an equilibrium where a small number of impressions are awarded to support reinforced learning.

It should be clear from the above illustrations how bidder A adapts its bidding gracefully if a new bidder C enters the competition at a high price point, or if bidder B for whatever reason changes its bidding (perhaps due to its ads' time-varying performance or as a result of control signal adjustments).

4.4. Preliminary algorithm and analysis

The basic functionality of the proposed exploration and exploitation algorithm is obtained by simply implementing Eqs. (24)–(25). However, this basic form of the algorithm leads to a numerically unstable nominal bid price $b_p(k) = \alpha(k)/\beta(k)$ and an unbounded bid uncertainty $b_u(k) = 1/\sqrt{\alpha(k)}$ if no impressions and events are observed for a long time. While the variance growing property is a key aspect of the algorithm, unstable and unbounded signals are not. Instability and unboundedness are easily avoided by introducing a small drift term in the state update equation to drive the state away from the origin. A sensible choice of this drift ensures that its impact is insignificant when a modest number of impressions and events are observed.

We propose a drift term that is proportional to the initial state according to $(1-\gamma)[\alpha_0, \beta_0]^T$, where γ is a constant less than but very close to one. Note that if the a priori event rate estimate is, say, 10^{-4} and the a priori uncertainty is 1 (i.e., 100%), then the initial state, by virtue of relationships (27)–(28) is $[\alpha_0, \beta_0]^T = [1/b_u^2(0), 1/(b_p(0)b_u^2(0))]^T = [1, 10^4]^T$. When this is multiplied by $1-\gamma$ and added to the state, the algorithm is protected against unbounded signals with an insignificant impact to the state evolution as soon as a modest number of impressions and events are observed. This leads to *Exploration and Exploitation Algorithm I*:

Algorithm 4.1 (*Exploration and Exploitation Algorithm I*). Given parameters γ, λ, initial state $[\alpha(0), \beta(0)]^T = [\alpha_0, \beta_0]^T$, and measurements $n_I(k), n_E(k)$; compute for $k = 1, 2, \ldots$

$$\begin{bmatrix} \alpha(k) \\ \beta(k) \end{bmatrix} = \gamma\lambda \begin{bmatrix} \alpha(k-1) \\ \beta(k-1) \end{bmatrix} + (1-\gamma) \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} + \begin{bmatrix} n_E(k) \\ n_I(k) \end{bmatrix} \tag{29}$$

$$B \sim \text{Gamma}\big(\alpha(k), \beta(k)\big) \tag{30}$$

Again, by design α_0, β_0, and γ are chosen so that the bias term can be neglected when the exploration and exploitation has reached steady state.

As usual, $b^\star$ denotes the highest competing bid price and n_I^{tot} the total number of available impressions in each time interval k. To analyze the dynamics of the algorithm we assume for simplicity that $b^\star$, n_I^{tot}, and p are constant. Difference equation (29), which appears to be linear, is actually a concealed nonlinear equation since $n_E(k)$ and $n_I(k)$ depend on the state $\alpha(k-1)$ and $\beta(k-1)$ in a nonlinear fashion. To understand the dependency, recall first the distributions of the three involved random variables:

$$N_E(k) \sim \text{Binomial}\,(n_I(k), p), \tag{31}$$

$$N_I(k) \sim \text{Binomial}\,\left(n_I^{tot}, \text{Prob}(B \geq b^\star)\right), \tag{32}$$

$$B \sim \text{Gamma}\,\left(\alpha(k-1), \beta(k-1)\right), \tag{33}$$

where $\text{Prob}(B \geq b^\star) := \text{Prob}(B \geq b^\star \,|\, \alpha(k-1), \beta(k-1))$. Note how the random variables are related: Event volume $n_E(k)$ depends on impression volume $n_I(k)$, which depends on $\text{Prob}(B \geq b^\star)$, which depends on the state $\alpha(k-1)$ and $\beta(k-1)$. Furthermore $\alpha(k-1)$ and $\beta(k-1)$ depend on $n_E(k-1)$ and $n_I(k-1)$ according to (29), which closes the feedback loop of a stochastic difference equation involving two probability mass functions and one probability density function.

Solving the second order, nonlinear, and stochastic difference equation defined by (29)–(32) is an open research problem; however, a lot can be learned by solving a deterministic and partially linearized approximation of this dynamical system. The solution to the approximate dynamics may also be validated in simulations and experiments.

Express $n_E(k)$ and $n_I(k)$ as their expected values plus additive noise

$$n_E(k) = n_I(k)p + \epsilon_E(k)$$
$$n_I(k) = n_I^{tot}(k)\text{Prob}(B \geq b^\star) + \epsilon_I(k),$$

where $\epsilon_E(k)$ and $\epsilon_I(k)$ satisfy $\text{E}(\epsilon_E(k)) = \text{E}(\epsilon_I(k)) = 0$, $\text{Var}(\epsilon_E(k)) = n_I(k)p(1-p)$ and $\text{Var}(\epsilon_I(k)) = n_I^{tot}(k)\text{Prob}(B \geq b^\star)\left(1 - \text{Prob}(B \geq b^\star)\right)$.

As a first approximation, replace $n_I(k)$ and $n_E(k)$ in (29)–(30) with their expected values. For sufficiently large values of λ and γ, we expect this approximation to be harmless. We obtain

$$\begin{bmatrix}\alpha(k)\\ \beta(k)\end{bmatrix} = \gamma\lambda\begin{bmatrix}\alpha(k-1)\\ \beta(k-1)\end{bmatrix} + (1-\gamma)\begin{bmatrix}\alpha_0\\ \beta_0\end{bmatrix} + \begin{bmatrix}p\\ 1\end{bmatrix} n_I^{tot}\text{Prob}(B \geq b^\star).$$

On the other hand, $\text{Prob}(B \geq b^\star) = \int_{b^\star}^{\infty} \beta^\alpha x^{\alpha-1} e^{-\beta x} dx / \Gamma(\alpha)$, see (20), where in the integral equation the abbreviated notation $\alpha := \alpha(k-1)$ and $\beta := \beta(k-1)$ is used to avoid clutter. Moreover, (21) asserts that $\text{Prob}(B \geq b^\star) = \text{Prob}(\tilde{B} \geq b^\star\beta)$ where $\tilde{B} \sim \text{Gamma}\,(\alpha, 1)$, and we recall from (26) that $\alpha(k-1)/\beta(k-1) = \hat{p}(k-1)$. It follows that

$$\text{Prob}(B \geq b^\star) = \int_{b^\star\alpha/\hat{p}}^{\infty} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} dx.$$

The event rate estimate $\hat{p}(k)$ contains estimation error, which we express $\hat{p}(k) = (1 + \epsilon_{\hat{p}}(k))p$. Linearization of $\text{Prob}(B \geq b^\star)$ at $\hat{p} = p$ results in

$$\text{Prob}(B \geq b^\star) = \int_{b^\star\alpha/p}^{\infty} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} dx + a\epsilon_{\hat{p}}(k-1) + \mathcal{O}(\epsilon_{\hat{p}}(k-1)^2),$$

where a is a linearization coefficient that depends on $b^\star/p$ and α. Our second approximation is to assume that the impact from $\epsilon_{\hat{p}}(k-1)$ is insignificant and that we can neglect both the linear and all higher order terms of $\epsilon_{\hat{p}}(k-1)$. It can be shown that the impact of $\epsilon_{\hat{p}}(k-1)$ decreases rapidly as $b^\star/p$ deviates from one.

The above approximations, $\epsilon_E(k) = \epsilon_I(k) = \epsilon_{\hat{p}}(k-1) = 0$, lead to two linearly dependent deterministic difference equations, which are far easier to analyze than the original higher-dimensional stochastic equations. In summary, we have arrived at the following approximate model:

$$\alpha(k) = \gamma\lambda\alpha(k-1) + (1-\gamma)\alpha_0 + pn_I^{tot} \int_{b^\star\alpha/p}^{\infty} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} dx \tag{34}$$

This equation does not have a closed form solution, but is well-behaved and can be analyzed conveniently using graphical techniques. We are interested in the fixed point and transient solutions of the equation.

4.5. Fixed point solutions

The special cases where $b^\star = 0$, $n_I^{tot} = 0$, or $p = 0$ are trivial to analyze starting with (29)–(32). The details are left to the reader.

The more common and interesting case applies when $b^\star$, n_I^{tot}, and p are non-zero. Begin the analysis by setting $\alpha(k) = \alpha(k-1) \equiv \alpha$. Then rewrite (34) as an equality between a unit-less linear left-hand equation and a unit-less nonlinear right-hand side as follows:

$$-\frac{(1-\gamma)\alpha_0}{pn_I^{tot}} + \frac{(1-\gamma\lambda)}{pn_I^{tot}}\alpha = \int_{b^\star\alpha/p}^{\infty} \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} dx. \tag{35}$$

The unit-less representation makes the results easier to interpret and allows us to develop a deeper intuition of the behavior of the algorithm that is not dependent on specific parameter values. It can be shown that this equation has a unique fixed point solution (there would be a second trivial solution at $\alpha = 0$ if $\gamma = 1$ since the right-hand side integral surprisingly converges to zero(!) as $\alpha \to 0$).

A great deal of insight about the fixed point solutions is gained by drawing the left- and right-hand sides of (35) separately as functions of α and identifying the intersecting point. This is done in Fig. 11 for $n_I^{tot} = 1000$, $\lambda = 0.975$, $\gamma = 0.99$, $b^\star = 0.02$, and $\alpha_0 = 2.5$. In many real applications, n_I^{tot} is much larger then 1000. But the chosen value makes it easier to understand the qualitative behavior of the fixed point solutions. Inspect (35) to appreciate how the fixed point solution depends on the value of n_I^{tot}. The left-hand side equation is inversely proportional to n_I^{tot}, while the right-hand side equation is independent of n_I^{tot}.

Each black curve in Fig. 11 represents the right-hand side of (35) for a specific value of the ratio $b^\star/p$. Three of the black curves have been tagged with "$b^\star/p = 0.9$", etc. Each red line represents a specific configuration of α_0, λ, γ, p, and n_I^{tot}. In fact, given the above prescribed values of α_0, λ, γ, and n_I^{tot}, the red lines differ only in terms of p. Two of the red lines have been tagged with "$p = 1.1b^\star$", etc. The intersection between a black curve and a red line for a specific p is marked with a green dot, representing the fixed point for a given configuration. Each fixed point is associated with a value of α and a '% Awarded Impressions', which is the probability

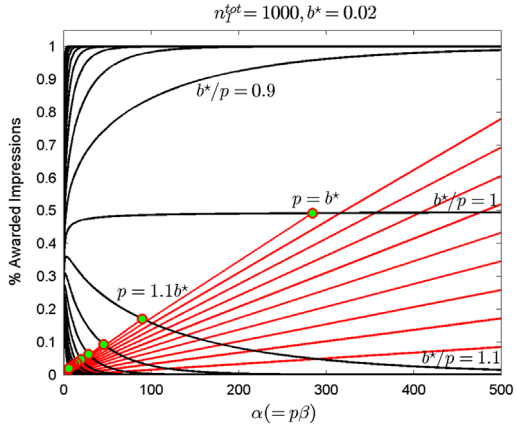

Fig. 11. Graphical representation of the fixed point solution of Heisenberg bidding based Exploration & Exploitation. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

the bidder wins an impression if the state has converged to the fixed point.

Note how the black curve as a function of α converges to an allocation of 0.5 when $b^\star/p=1$, but how it rapidly approaches 1 (or 0) when $b^\star/p$ is only slightly less than (or larger than) one. The curves reach 1 or 0 only at the limit $\alpha\to\infty$, so the fixed point allocation which is defined by the intersecting point is close to, but never identically, 0 or 1. In other words, if p is slightly less than $b^\star$, then a very small (but non-zero) allocation of impressions is awarded to the bidder. If p is slightly larger than $b^\star$, then nearly all impressions are allocated to the bidder. Furthermore, as p, n_I^{tot}, λ, or γ increase; the slope of the red line decreases and the intersecting point with the black curves happens at a larger value of α with a percentage impression allocation closer to zero or one depending on if p is smaller or larger than $b^\star$. This, in a nutshell, illustrates the exploration–exploitation trade-off. The vast majority of impressions are awarded to the highest performing bidder, but a small number is awarded to lower performers for the purpose of an ongoing learning and where the impression allocation adapts to time-varying values of p, n_I^{tot}, λ, γ, and $b^\star$ with no need of human intervention.

Visually inspect the red lines and black curves in the figure, and compare them with (35) to understand how the fixed point depends on the known and unknown parameters. This allows you to develop a good understanding of how the intersecting fixed point moves as the parameter values change.

Further insight is gained by plotting the fixed point solutions in terms of the expected number of awarded impressions n_I^{tot} Prob $(B\geq b^\star)$ versus $p/b^\star$. This is done for $n_I^{tot}=10,000$ and three different values of $b^\star$ in Fig. 12. The right subplot shows in greater detail the expected number of impressions for low-performing bidders ($p/b^\star<1$). The take-away is, again, that a small fraction of the impressions are awarded to the bidder if $p<b^\star$, and that the expected number of impressions grows with $p<b^\star$ but remains small until the ratio is very close to one. For example, if $p=b^\star/2$ then we may expect only between about 20 and 60 impressions out of a total of 10,000 to be awarded to fuel the ongoing reinforced learning/exploration — a number that grows to a few hundred impressions, at most, if $p=0.9b^\star$, but rapidly to approximately 10,000 when $p/b^\star$ exceeds one. The number of awarded impressions increases as $b^\star$ decreases (for a constant ratio of $p/b^\star$) as the result of the greater challenge of estimating a smaller event rate.

Fig. 13 illustrates the above behavior for $n_I^{tot}=100,000$ (with $\lambda=0.975$, $\gamma=0.99$, and $\alpha_0=2.5$) and shows how the number of impressions awarded to a bidder with $p<b^\star$ grows sub-linearly as a function of n_I^{tot}. Indeed, while n_I^{tot} differs by a factor of ten between the two examples, the number of impressions awarded to a low-performing bidder increases by less than a factor of two. Practically, this means the algorithm requires less custom-tuning than many conventional algorithms.

As a final remark, the input measurement to Algorithm 4.1 for a specific bidder consists exclusively of the impression and event counts for the bidder itself. Thus it does not rely on the total available impression volume n_I^{tot}, or the measurements or bids of other bidders. The decentralized nature of the algorithm where no information is exchanged among the different bidders besides what is encoded in a bidder's own impression and event measurements makes the system extremely scalable.

4.6. Transient behavior

Now consider the transient behavior of (34). The dynamics is nonlinear without a closed form solution, but is described by a first order difference equation and can therefore be analyzed graphically using a cobweb plot [10]. The cobweb plot draws the trajectory of $\alpha(k)$ by marking and connecting the coordinates $(\alpha(0),\alpha(1))$, $(\alpha(1),\alpha(2))$, etc. It is an excellent tool for gaining intuition on how a general nonlinear, but low-dimensional, discrete-time dynamical system evolves for different parameter configurations and initial conditions.

For example, Fig. 14 shows the transient behavior of $\alpha(k)$ when $\alpha_0=2.5$, $\lambda=0.975$, $\gamma=0.99$, $b^\star=0.05$, and $n_I^{tot}=1,000,000$ for two different values of p. The top and bottom rows correspond to $p=0.01$ and $p=0.014$, respectively. The left column shows the cobweb plots in which the fixed point solutions are given by the (unique) points along the red curve where $\alpha(k)=\alpha(k+1)$. The center column shows the time series plot of $\alpha(k)$, and the right column shows the time-series plot of $b_u(k)$, where $b_u(k)=1/\sqrt{\alpha(k)}$.

In the case of $p=0.01$ (top row), the initial value of α is slightly smaller than the fixed point and instantaneously is mapped to a value $\alpha(1)$, which is larger than the fixed point. However, the overshoot is modest and it does not take long until $\alpha(k)$ has decreased to a value very close to the fixed point. Recall that the fixed point represents a value where the trade-off between exploration and exploitation has reached an equilibrium. For a scenario where $p<b^\star$ this corresponds to a state where a very small number of impressions are awarded to our bidder. The good behavior is confirmed in all three subplots.

In the case of $p=0.014$ (bottom row), the mapping from α_0 to $\alpha(1)$ is more dramatic than in the first example. While the initial values are identical in the two examples, the mappings defined by (34), and the fixed points, are different resulting in a very different transient behavior. The overshoot of $\alpha(1)$ is much larger and the time it takes for $\alpha(k)$ to get close to the fixed point is much longer. The long convergence time is evident in all three plots, but the implication is best understood in the $b_u(k)$ plot. The small value of

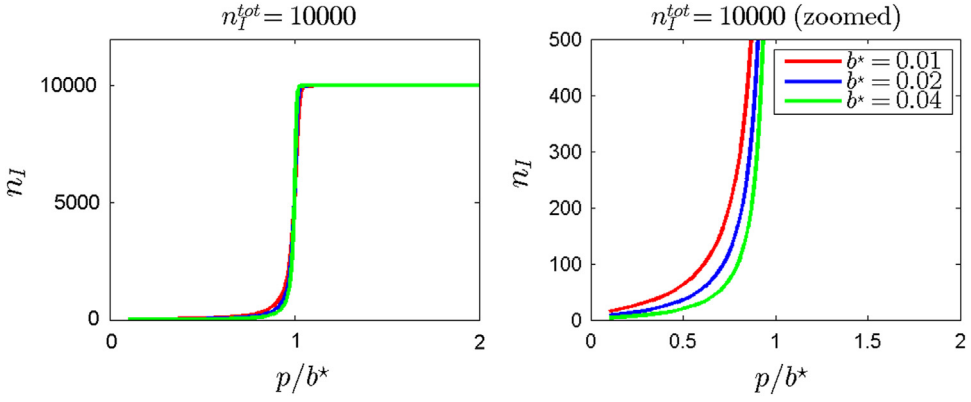

Fig. 12. Graphical representation of the fixed point solution of Heisenberg bidding based Exploration & Exploitation as a function of $p/b^\star$ for $n_I^{tot}=10,000$.

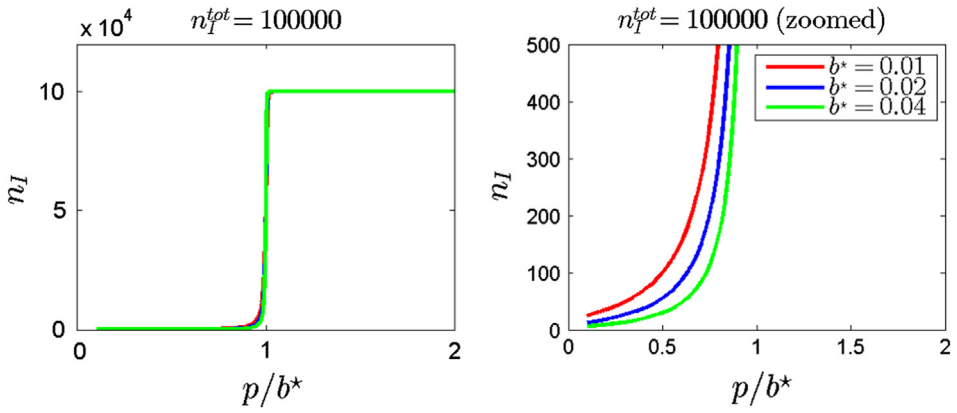

Fig. 13. Graphical representation of the fixed point solution of Heisenberg bidding based Exploration & Exploitation as a function of $p/b^\star$ for $n_I^{tot} = 100,000$.

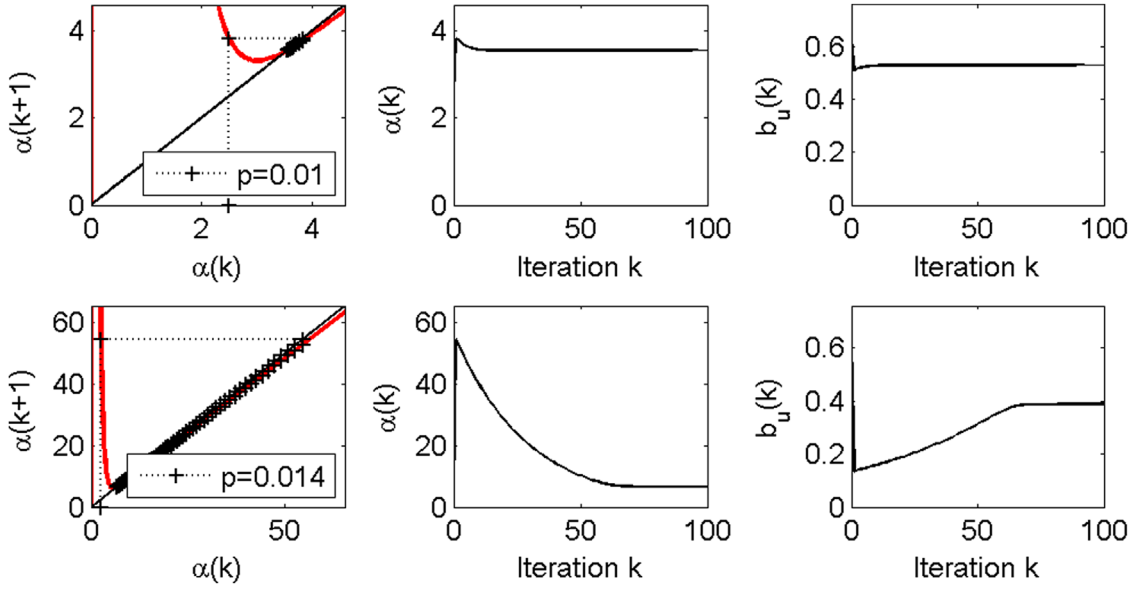

Fig. 14. Two representative examples of cobweb and time-series plots for Heisenberg Bidding-based Exploration and Exploitation Algorithm 1 ($\alpha_0 = 2.5$, $\lambda = 0.975$, $\gamma = 0.99$, $b^\star = 0.05$, and $n_I^{tot} = 10^6$). (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

$b_u(k)$ for many hours is indicative of a spiky bid distribution, which in this case where $p < b^\star$ typically means no impressions at all are awarded for 50 or more iterations. This is an undesirable behavior since it means there is no exploration. Effectively, the bidder is blindfolded and learns nothing about its performance.

With help of the cobweb plot we can understand the different behaviors and thereafter devise an approach on how to prevent a slow convergence. The red curves in the cobweb plots (left subplots in Fig. 14) draw the dynamics defined by (34), which consists of a linear and a nonlinear (integral) term. By inspection of the equation, we verify that the integral term tends to be very large for small values of α (except for extremely small values), but converges to zero when $\alpha \to \infty$. On the other hand, the linear term converges to zero when $\alpha \to 0$ and grows unboundedly as $\alpha \to \infty$. Consequently, the left segment of the curve is dominated by the integral term and the right segment by the linear term. The red curve is a function of several unknown parameters (p, n_I^{tot}, and $b^\star$) in addition to parameters set by us (γ, λ, and α_0), and we do not know a priori what the fixed point is or what is a good value of α_0 to prevent slow convergence.

There are two factors causing the slow convergence in the second example. First, the red curve grows rapidly as $\alpha(k) \to 0$ and as a result $\alpha(k+1)$ is very large whenever $\alpha(k)$ is much smaller than its fixed point solution. Preventing this is the topic of next subsection. Secondly, the asymptote of the red curve for large values of $\alpha(k)$ is very close to the black line tilted at 45°, which means $|\alpha(k+1) - \alpha(k)|$ is very small whenever $\alpha(k)$ is large. We may reduce the slow convergence for large α by reducing the slope of the asymptote, and since the slope equals $\gamma\lambda$ a natural choice is to reduce λ.

However, as a word of caution, faster convergence is desired, but there is no free lunch. A smaller value of λ leads to a higher expected percentage of impressions being awarded to a bidder with $p < b^\star$ (see Section 4.5), hence is associated with a larger exploration cost. Furthermore, a too small value of λ leads to a very complex dynamics involving for example limit cycles, chaos, or instability.

4.7. Enhanced algorithm

One important culprit of slow convergence as shown in the bottom row of Fig. 14 is that $\alpha(k+1)$ may overshoot the fixed point by a lot if $\alpha(k)$ is much smaller than the fixed point. With a large overshoot, the convergence down to the fixed point is slow since the incremental steps $\alpha(k+1) \to \alpha(k+2) \to \cdots$ are tiny. The physical interpretation of the behavior is explained with help of Fig. 10, which shows an example where $p < b^\star$. Indeed, if $\alpha(k)$ is allowed to increase by a large amount in a single update, then the bid

distribution instantaneously becomes spiky (recall that $b_u(k) = 1/\sqrt{\alpha(k)}$). On the other hand, if $p \ll b^\star$ then the spiky distribution means approximately no impressions are awarded and no events generated.

To reduce the risk of slow convergence without introducing additional dynamics, which would make the dynamical analysis more difficult, we propose one simple modification of Algorithm 4.1. We recommend imposing an upper bound in relative sense on how much $\alpha(k)$ or $\beta(k)$ are allowed to increase from time k to $k+1$. This is achieved by scaling $n_E(k)$ and $n_I(k)$ with a common multiplicative scaling factor before updating $\alpha(k)$ and $\beta(k)$. In particular, introduce a parameter $\rho > 1$ specifying by what relative amount $\beta(k)$ is allowed to change from k to $k+1$. This mechanism improves the transient behavior of the estimator and is summarized in Algorithm 4.2.

Algorithm 4.2 (*Exploration and Exploitation Algorithm 2*). Given parameters γ, λ, ρ, initial state $[\alpha(0), \beta(0)]^T = [\alpha_0, \beta_0]^T$, and measurements $n_I(k), n_E(k)$; compute for $k = 1, 2, \ldots$

$$\tilde{n}_I(k) = \min\big(n_I(k), \rho\beta_i(k)\big)$$

$$\tilde{n}_C(k) = \begin{cases} \dfrac{\tilde{n}_I(k)}{n_I(k)} n_C(k), & \text{if } n_I(k) > \tilde{n}_I(k) \\ n_C(k), & \text{otherwise} \end{cases}$$

$$\begin{bmatrix} \alpha(k) \\ \beta(k) \end{bmatrix} = \gamma\lambda \begin{bmatrix} \alpha(k-1) \\ \beta(k-1) \end{bmatrix} + (1-\gamma) \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} + \begin{bmatrix} \tilde{n}_C(k) \\ \tilde{n}_I(k) \end{bmatrix}$$

$$B \sim \text{Gamma}\big(\alpha(k), \beta(k)\big)$$

Fig. 15 shows how the transient behavior of the second example in Fig. 14 is improved by the added mechanism. Besides the bound on how much $\beta(k)$ may change, we have here also reduced the value of λ. In particular, in this example we use $\gamma = 0.99$, $b^\star = 0.05$, $\alpha_0 = 2.5$, $n_I^{tot} = 1,000,000$, $\rho = 4$, and $\lambda = 0.8$. The plots speak for themselves, but compare the figure with the dynamics described by (34) to better understand how parameter variations may change the behavior.

4.8. Experimental results

Heisenberg bidding based exploration and exploitation [18] is versatile and used in several applications beyond the advertisement event rate estimation problem described in Section 2. We shall give a few examples, but begin with the earliest application and discuss this example in most detail.

The first application of Heisenberg bidding-based exploration and exploitation was to automate and optimize the content promotions on www.aol.com, a large web portal receiving millions of page views daily and functioning as the entry point for many Internet users consuming content within AOL's owned and operated properties (including familiar sites such as Huffington Post and TechCrunch).

The algorithm maximizes the number of clicks in the, so-called, Dynamic Lead, which is located in the upper left corner of the web page and typically consists of 40 promotional slots. The Dynamic Lead shows only one slot at a time, starting with slot 1 upon page load and stepping forward to slots 2, 3, etc., every few seconds, and eventually returning to slot 1 after slot 40. The optimization is achieved by performing Heisenberg bidding based exploration and exploitation on a content pool curated by editors. Individual contents are added (published) to or removed (archived) from the pool at the discretion of editors.

Each time a user loads www.aol.com, a random final bid is generated for each available content based on its nominal bid price (the optimal Bayesian event rate estimate) and bid uncertainty (the posterior relative standard deviation of the event rate estimate). The event in this application is a "click," hence the event rate is a *click-through-rate* (CTR). A market clearing mechanism sorts the final bids in decaying order and identifies the content promotions associated to the first 40 bids in the list. The first content promotion is served in slot 1, the second in slot 2, and all the way to slot 40. Consistent with the fixed point solution determined in Section 4.5, the promotion served in slot 1 with the highest final bid price is almost always the promotion with the highest estimated CTR. The impression and click counts are recorded and fed to the algorithm every five minutes and are used to update the nominal bid price and bid uncertainty for each bidder. Note that an impression for a content promotion is registered only if the user stays on the portal web page long enough for the slot with the content promotion to be visible.

The initial state $[\alpha_0, \beta_0]^T$ was selected using empirical Bayesian modeling, where we fitted a gamma distribution to a histogram of CTR's computed from historical data from more than 1000 content promotions. This is a one-time simple fitting exercise for each new use case of the algorithm, and the result for the Dynamic Lead on www.aol.com is shown in Fig. 16. Note the good fit between historical CTR estimates and the gamma distribution, where the end result is $\alpha_0 = 2.44$ and $\beta_0 = 263$. It is not very important to have a good fit since α_0, β_0 are only initial values of the state, but a good choice of α_0, β_0 means the content initially beats the competition just often enough to support the exploration and reinforced learning of the CTR of the content without resulting in a high opportunity cost in the event the content turns out to be low-performing.

A value $\lambda = 0.85$ was chosen by considering the typical life span of a promotion, and by doing off-line analysis of historical data to assess the typical time-variability of CTR rates. The criteria used to select λ were primarily responsiveness, bias (due to trends in p), and exploration cost.

On the other hand, $\gamma = 0.99$ was selected to provide a safeguard that does not impact the behavior under normal operating conditions. It was chosen primarily based on a desired steady-state

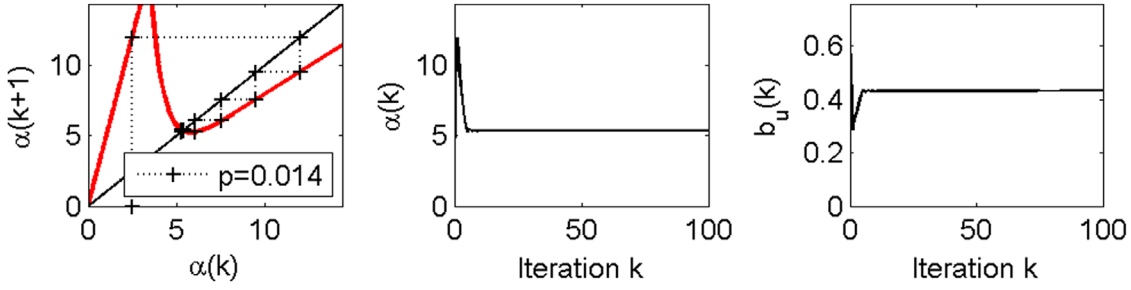

Fig. 15. Representative example of cobweb and time-series plots for Heisenberg Bidding-based Exploration and Exploitation Algorithm 2 ($\alpha_0 = 2.5$, $\lambda = 0.8$, $\gamma = 0.99$, $b^\star = 0.05$, $n_I^{tot} = 10^6$, $p = 0.014$, and $\rho = 4$).

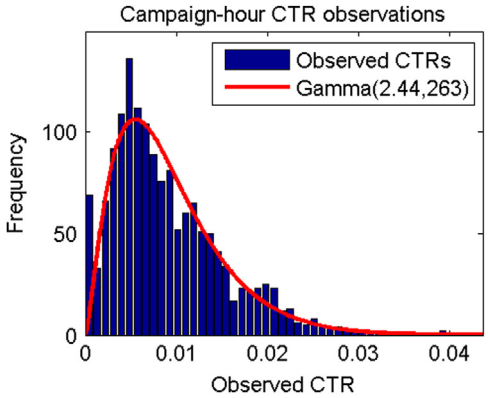

Fig. 16. Empirical Bayesian modeling of the prior distribution of the CTR and the corresponding initial state ($\alpha_0 = 2.44$ and $\beta_0 = 263$) of the exploration and exploitation algorithm.

behavior of b_p and b_u in the extreme case of no impression or event observations. Note that the steady-state solution in this situation is $b_p = \alpha_0/\beta_0$ and $b_u = \sqrt{(1-\gamma\lambda)/((1-\gamma)\alpha_0)}$.

The Heisenberg bidding based exploration and exploitation algorithm has been used for many months. Fig. 17 shows CTR estimates and impression allocation for a representative one hour period. The pool in this hour consists of 138 content promotions. The bars in the top-most subplot display the estimated CTR for all promotions ordered according to decreasing values. Since this is experimental data there is no ground truth, so we use the CTR estimate as a proxy for the true CTR. The second and third subplots (in linear and logarithmic scale, respectively) present the number of impressions each promotions recorded in this hour period. Notice that the highest performing promotion was awarded most impressions (around 260,000) as the result of almost always being selected for the first slot in the Dynamic Lead rotation.

Observe also that promotions 1 to (approximately) 45 were awarded dramatically many more impressions than promotions 46–138, but all promotions were awarded at least a small number of impressions. This relative impression allocation demonstrates that the exploration–exploitation trade-off is consistent with the theoretical results in Figs. 11–13.

A different perspective of the experimental exploration versus exploitation trade-off and the transient behavior is shown in Fig. 18. The plots compare the impression allocation and CTR estimation of two sample content promotions. Each column of plots corresponds to one promotion. The top row of the plot displays the number of recorded impressions for each promotion in each five-minute time interval. In the bottom row of the plot the red markers indicate the observed CTR for each five-minute interval and the green curve represents the estimated CTR computed recursively by the experimental exploration–exploitation algorithm.

We see that the high-performing promotion recorded massively more impressions than the low-performer and that the CTR estimation as a result is easy. Furthermore, it should be noted that the high-performing promotion is subject to "burn-out" — the decaying CTR over time. The observed and estimated CTR dropped about 50% over the course of 24 h.

The low-performing promotion, on the other hand, recorded hundreds of impressions in the first 15–30 min, but thereafter only a very small number of impressions in each five-minute interval. The small number of impressions supports reinforced learning, ensuring the estimator has a chance to detect time-varying changes to the underlying true CTR, but in a cost-efficient manner since the opportunity cost of the trickle of impressions is almost zero. Realize that the CTR estimation for the low-

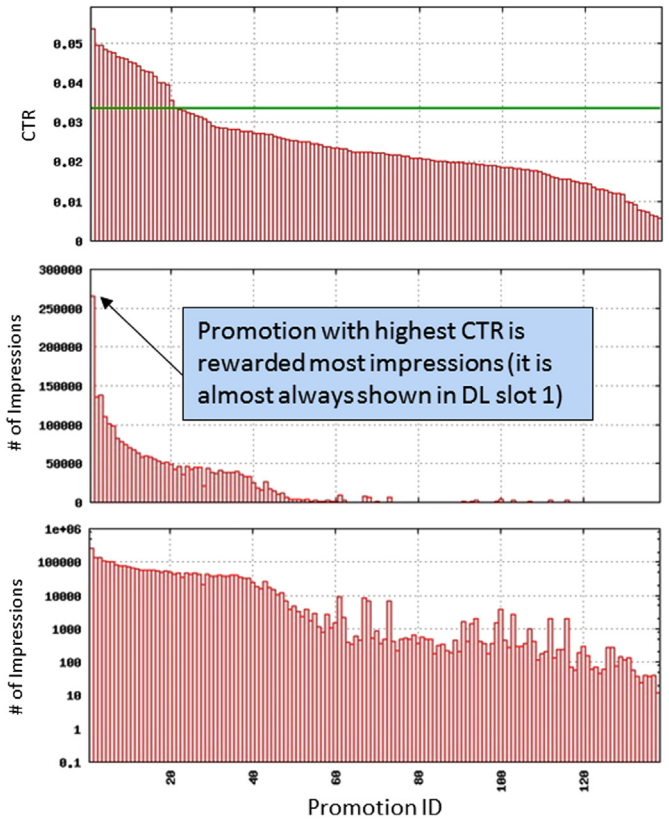

Fig. 17. Bar charts showing the CTR estimates and the number of impressions for each available content promotion in a representative hour period.

performing promotion is more difficult since it is based on a tiny number of impressions and clicks. The observed CTR values, per 5-min time interval, in the bottom right plot are usually zero, but now and then somewhere between 1% and 9%. In spite of this noisy data, the estimated CTR is robust and appears trustworthy.

Finally, let us mention briefly a few examples of Heisenberg bidding-based exploration and exploitation unrelated to the content promotions on www.aol.com. For instance, it is implemented as the brain of AOL's One Creative™ product, where it is used to determine which version of an ad (from the same advertiser) is most appealing to a user. The ads may have different headline text, poster frame photos, background color, offer detail, and so forth. The optimization objective is to determine the most efficient way a specific ad configuration leads to the highest response rate for each user segment. Another application of the proposed exploration and exploitation algorithm is to find out which poster frame from a set of candidates to use in the promotion of an online video. This application is a feature of AOL's VideoLearn™ system. Yet another application under development is to use the algorithm as the brain of CommerceLearn™, which intends to optimize product recommendations. A common element of all these applications is that they require a scalable solution and that the exploration and exploitation capability is merely one building block of the system, and is required to work in harmony with a separate and interacting feedback control system used for some form of optimization and/or supply chain management [22].

5. Concluding remarks

In this paper, we have discussed a relatively new application of feedback control related to online advertising. Online advertising is a growing industry with plenty of interesting and challenging

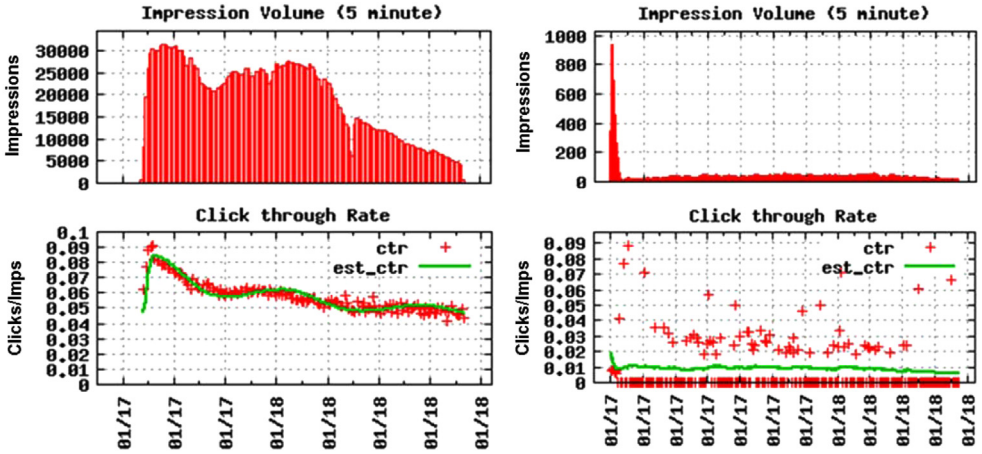

Fig. 18. Two sample content promotions. Top: impression volume per 5 min; bottom: observed clicks/impressions and estimated CTR ($=b_p$); left: high-performing promotion; right: low-performing promotion. (For interpretation of the references to color in this figure, the reader is referred to the web version of this paper.)

dynamics, statistics, and computer science problems. Applying scalable optimization to these problems requires advanced feedback control.

A plant model is derived from first principle to highlight some of the challenges a control engineer faces in this new and exciting area. It is by no means the perfect model, and it does involve simplifying assumptions. However, our intent is to make the reader appreciate the unique challenges of designing control systems for online advertising, and to make it clear in what way estimation errors, competing bid prices, and volatile Internet traffic influence the behavior of the plant, as perceived by the control system.

Several of the challenges relate to the discontinuous nature of the plant, and we propose randomized bidding as a mitigation to many of them. Bid randomization effectively removes discontinuities, reduces the impact of process variations and noise, and enhances the dynamic interaction among bidding agents. It also creates a fixed point solution to control problems that otherwise do not have a solution, and much more. Bid randomization is useful to solve the problem of exploration and exploitation, and techniques from nonlinear dynamics can be used to analyze the system behavior, including the transient and steady state solution.

Acknowledgments

The author is grateful to Mohammad Ziaemohseni, Qian Sang, Deborah Richards, and Jiaxing Guo for their helpful feedback on the manuscript.

Appendix A. Proof sketch of optimality of $\mathbf{b_i = u^{opt} p_i}$

The purpose of this paper is not to prove optimality of bidding strategy (8)–(10). A rigorous proof of this claim is therefore not offered; however, to make the result plausible, we outline the idea of a proof and leave the details for the reader to work out.

Define decision variable $x_i(b_i, a_i) = a_i \mathbb{I}_{\{b_i \geq b_i^\star\}}$ and rewrite the ROI constraint $r \geq u_{ROI}^{ref}$ in terms of v and c. Since $r = v/c$, the constraint is $v/c \geq u_{ROI}^{ref}$, or $u_{ROI}^{ref} c \leq v$. We can then rewrite the optimization problem (5)–(7) as a Linear Program

$$\max_{\substack{0 \leq x_i \leq 1 \\ i=1,2,\ldots}} \sum_i v_E p_i n_{l,i}^{tot} x_i$$

subject to

$$\sum_i b_i^\star n_{l,i}^{tot} x_i \leq u_{cost}^{ref}$$

$$\sum_i \left(u_{ROI}^{ref} b_i^\star - v_E p_i \right) n_{l,i}^{tot} x_i \leq 0.$$

Partition the segments based on the values of p_i and $b_i^\star$. Denote the set of all segments Ω. Let $\Omega_1 = \{i | b_i^\star = p_i = 0, i \in \Omega\}$, $\Omega_2 = \{i | b_i^\star = 0, p_i > 0, i \in \Omega\}$, $\Omega_3 = \{i | b_i^\star > 0, p_i = 0, i \in \Omega\}$, and $\Omega_4 = \{i | b_i^\star > 0, p_i > 0, i \in \Omega\}$.

Impressions awarded from segments in Ω_1 do not contribute to neither the objective function ($p_i = 0 \Rightarrow v_i = 0$) nor toward constraints ($b_i^\star = 0 \Rightarrow c_i = 0$, etc.), hence x_i can be chosen arbitrarily. Our choice is $x_i = 1$ if $i \in \Omega_1$.

Impressions awarded from segments in Ω_2 contribute with a positive value to the objective function, and effectively relax the second constraint, but with no added cost, hence x_i should be chosen as large as possible; i.e., $x_i = 1$ if $i \in \Omega_2$.

Impressions awarded from segments in Ω_3 do not contribute to the objective function, but add cost, hence x_i should be chosen as small as possible; i.e., $x_i = 0$ if $i \in \Omega_3$.

Impressions from segments in Ω_4 require further analysis. We have

$$\max_{\substack{0 \leq x_i \leq 1 \\ i=1,2,\ldots}} \sum_{i \in \Omega_2} v_E p_i n_{l,i}^{tot} + \sum_{i \in \Omega_4} v_E p_i n_{l,i}^{tot} x_i$$

subject to

$$\sum_{i \in \Omega_4} b_i^\star n_{l,i}^{tot} x_i \leq u_{cost}^{ref} - \sum_{i \in \Omega_2} b_i^\star n_{l,i}^{tot}$$

$$\sum_{i \in \Omega_4} \left(u_{ROI}^{ref} b_i^\star - v_E p_i \right) n_{l,i}^{tot} x_i \leq \sum_{i \in \Omega_2} v_E p_i n_{l,i}^{tot}.$$

The first sum in the objective function is independent of x_i and can therefore be ignored. This same sum appears on the right-hand side of the second constraint, and since it is constant it can be treated as a fixed relaxation of the second constraint.

Assume without loss of generality that the segments in Ω_4 are indexed so that $0 < b_1^\star/p_1 < b_2^\star/p_2 < \cdots$. Since the ROI of impressions in segment i equals $v_E p_i / b_i^\star$, it follows that the ROI associated with each segment decreases with segment index i; i.e., $r_1 > r_2 > \cdots$. It can then be argued, and proven using standard techniques from linear optimization, that impressions from segment 1 should be chosen over impressions from segment 2, etc. The constraints dictate how many of the segments are of interest to the bidder. The optimal solution of the Linear Program turns out to be $x_1 = x_2 = \cdots = x_{i'-1} = 1$, $x_{i'} = a_{i'}$, and $x_{i'+1} = x_{i'+2} = 0$, where i' and $a_{i'}$ are selected to make one of the constraints binding (if

possible). They are given by (10). If the constraints cannot bind for any x_i's, then the optimal solution is $x_i=1$ for all i.

Since $x_i(b_i,a_i)=a_i\mathbb{I}_{\{b_i\geq b_i^\star\}}$ it follows that optimality requires $a_i=\mathbb{I}_{\{b_i\geq b_i^\star\}}=1$ for $i=1,\ldots,i'-1$. This is possible only if $a_1=\cdots=a_{i'-1}=1$ and $b_i\geq b_i^\star$ for $i=1,\ldots,i'$. Optimality furthermore requires that $a_{i'}$ is given by (10) and $\mathbb{I}_{\{b_{i'}\geq b_{i'}^\star\}}=1$, which is only possible if $b_{i'}\geq b_{i'}^\star$. Finally, optimality requires $a_i\mathbb{I}_{\{b_i\geq b_i^\star\}}=0$, which is possible if $a_i=1$ and $b_i<b_i^\star$ for $i=i'+1,i'+2,\ldots$.

It is straightforward to verify that the optimality conditions on b_i for all Ω_4 impressions are satisfied if $b_i=u^{opt}p_i$, where u^{opt} is given by (8). It is also trivial to confirm that this bidding strategy is consistent with the optimal bidding on impressions from segments in Ω_1, Ω_2, and Ω_3.

Appendix B. Proof of monotonicity of $c(u)$, $v(u)$, and $r(u)$

First insert bidding strategy $b_i=up_i$ in (2) and (3) to obtain $c(u)=\sum_i a_i b_i^\star\mathbb{I}_{\{up_i\geq b_i^\star\}}n_{l,i}^{tot}$ and $v(u)=\sum_i a_i p_i v_E\mathbb{I}_{\{up_i\geq b_i^\star\}}n_{l,i}^{tot}$.

B.1. $c(u)$ and $v(u)$ are non-decreasing

Since $a_i,b_i^\star,n_{l,i}^{tot},p_i,v_E\geq 0$ while $\mathbb{I}_{\{up_i\geq b_i^\star\}}$ is a step function with positive step, it follows that $c(u)$ and $v(u)$ are sums of non-decreasing functions, which immediately proves that $c(u)$ and $v(u)$ are non-decreasing.

B.2. $r(u)$ is non-increasing

Recall that $u\geq 0$. Clearly, segments with both $p_i=0$ and $b_i^\star>0$ have no impact at all on $r(u)$, while segments with both $p_i\geq 0$ and $b_i^\star=0$ contribute with a vertical shift that does not impact the monotonicity of the function. Since the above special cases of segments do not influence the monotonicity they can be disregarded as we proceed with the proof. Consider therefore only segments where $p_i>0$ and $b_i^\star>0$. Moreover, assume without loss of generality $0<b_1^\star/p_1<b_2^\star/p_2<\cdots$.

As a direct consequence of how $v(u)$ and $c(u)$ are defined, it follows that $r(u)=v(u)/c(u)$ only changes value at the discrete points where one of the indicator functions $\mathbb{I}_{\{up_i\geq b_i^\star\}}$ changes value; i.e., at each value of u for which $u=b_i^\star/p_i$ for some i. Denote this sequence $u_k=b_k^\star/p_k$ for $k=1,2,\ldots$, which by assumption satisfy $u_k\leq u_{k+1}$. Therefore, $r(u)$ is non-increasing if and only if $r(u_k)$ is a non-increasing sequence. Replacing u with u_k in (2) and (3) gives us $v(u_k)=\sum_{i=1}^k a_i p_i v_E n_{l,i}^{tot}$ and $c(u_k)=\sum_{i=1}^k a_i b_i^\star n_{l,i}^{tot}$, whereafter we can derive a recursive formula for $r(u_k)$ as follows:

$$\begin{aligned}
r(u_{k+1})&=\frac{v(u_{k+1})}{c(u_{k+1})}=\frac{v(u_k)+a_{k+1}p_{k+1}v_E n_{l,k+1}^{tot}}{c(u_{k+1})}\\
&=\frac{c(u_k)}{c(u_{k+1})}\frac{v(u_k)}{c(u_k)}+\frac{a_{k+1}b_{k+1}^\star n_{l,k+1}^{tot}}{c(u_{k+1})}\frac{a_{k+1}p_{k+1}v_E n_{l,k+1}^{tot}}{a_{k+1}b_{k+1}^\star n_{l,k+1}^{tot}}\\
&=\frac{c(u_k)}{c(u_{k+1})}r(u_k)+\frac{c(u_{k+1})-c(u_k)}{c(u_{k+1})}\frac{p_{k+1}v_E}{b_{k+1}^\star}\\
&=\alpha(k)r(u_k)+(1-\alpha(k))\frac{p_{k+1}v_E}{b_{k+1}^\star},
\end{aligned}\tag{B.1}$$

where $\alpha(k)=c(u_k)/c(u_{k+1})$, which satisfies $0<\alpha(k)\leq 1$ since $c(u_k)>0$ is a non-decreasing function. In other words, $r(u_{k+1})$ is bounded between $r(u_k)$ and $p_{k+1}v_E/b_{k+1}^\star$. It remains to prove that $p_{k+1}v_E/b_{k+1}^\star\leq r(u_k)$ is true for all k.

Next step involves mathematical induction. Suppose $p_{l+1}v_E/b_{l+1}^\star\leq r(u_l)$ for some l. It follows from (B.1) that $p_{l+1}v_E/b_{l+1}^\star\leq r(u_{l+1})\leq r(u_l)$. However, since $0<b_1^\star/p_1\leq b_2^\star/p_2\leq\cdots$ we have that $b_{l+1}^\star/p_{l+1}\leq b_{l+2}^\star/p_{l+2}$. Clearly, $p_{l+2}v_E/b_{l+2}^\star\leq p_{l+1}v_E/b_{l+1}^\star$, hence $p_{l+2}v_E/b_{l+2}^\star\leq r(u_{l+1})$, which means the induction assumption $p_{k+1}v_E/b_{k+1}^\star\leq r(u_k)$ holds for $k=l,l+1,\ldots$, and $r(u_l)\geq r(u_{l+1})\geq\cdots$.

To complete the proof, we confirm the assumption holds for $l=1$: We have $r(u_1)=v(u_1)/c(u_1)=a_1p_1v_En_{l,1}^{tot}/a_1b_1^\star n_{l,1}^{tot}=p_1v_E/b_1^\star\leq p_2v_E/b_2^\star$. Hence, we have proven by induction that $p_{k+1}v_E/b_{k+1}^\star\leq r(u_k)$ and $r(u_{k+1})\leq r(u_k)$ for $k=1,2,\ldots$, which completes the proof that $r(u)$ is a non-increasing function.

Note that the result is based on the bidding strategy $b_i=up_i$, hence monotonicity of $r(u)$ is not guaranteed for a bidding strategy $b_i=u\hat{p}_i$, where we only know that $\hat{p}_i\approx p_i$.

References

[1] S. Agrawal, N. Goyal, Analysis of Thompson sampling for the multi-armed bandit problem, in: Proceedings of the 25th Annual Conference on Learning Theory, 2012, pp. 39.1–39.26.
[2] S. Agrawal, N. Goyal, Further optimal regret bounds for Thompson sampling, in: Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, 2013, pp. 99–107.
[3] K.J. Åström, B. Wittenmark, Adaptive Control, 2nd ed., Prentice Hall, Reading, Massachusetts, 1994.
[4] J.-Y. Audibert, S. Bubeck, Minimax policies for adversarial and stochastic bandits, in: Conference on Learning Theory (COLT), 2009.
[5] P. Auer, N. Cesa-Bianchi, P. Fisher, Finite-time analysis of multiarmed bandit problem, Mach. Learn. 47 (2–3) (2002) 235–256.
[6] T. Basar, G.J. Olsder, Dynamic Noncooperative Game Theory, 2nd ed., SIAM, Philadelphia, PA, 1999.
[7] J.O. Berger, Statistical Decision Theory and Bayesian Analysis, 2nd ed., Springer-Verlag, New York, 1985.
[8] S. Bubeck, C.-Y. Liu, Prior-free and prior-dependent regret bounds for Thompson sampling, in: Advances in Neural Information Processing Systems 26 (NIPS), 2013, pp. 638–646.
[9] G. Casella, R.L. Berger, Statistical Inference, 2nd ed., Duxbury, Belmont, California, 2001.
[10] R.L. Devaney, An Introduction to Chaotic Dynamical Systems, 2nd ed., Westview, Boulder, Colorado, 2003.
[11] eMarketer, Last accesses January 6, 2016, Advertisers will spend nearly $600 billion worldwide in 2015 ⟨http://www.emarketer.com/Article/Advertisers-Will-Spend-Nearly-600-Billion-Worldwide-2015/1011691⟩.
[12] J. Ferber, S. Ferber, S. Kretsinger, R. Luenberger, D. Luenberger, Optimal Internet Ad Placement, United States Patent 7,822,636, USPTO, 2010.
[13] D. Fudenberg, J. Tirole, Game Theory, The MIT Press, Cambridge, Massachusetts, 1991.
[14] A. Garivier, O. Cappé, The KL-UCB algorithm for bounded stochastic bandits and beyond, in: Conference on Learning Theory (COLT), 2011.
[15] J. Gittins, K. Glazebrook, R. Weber, Multi-armed Bandit Allocation Indices, 2nd ed., Wiley, Chichester, United Kingdom, 2011.
[16] A. Gopalan, S. Mannor, Y. Mansour, Thompson sampling for complex online problems, in: Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 100–108.
[17] N. Karlsson, Systems and Methods for Controlling Bidding for Online Advertising Campaigns, United States Patent Application 20100262497, USPTO, 2009.
[18] N. Karlsson, Systems and Methods for Advertisement and Content Optimization over an Electronic Network, United States Patent Application 13/734,587, USPTO, 2013.
[19] N. Karlsson, Adaptive control using Heisenberg bidding, in: Proceedings of the 2014 American Control Conference, 2014, pp. 1304–1309.
[20] N. Karlsson, J. Zhang, Applications of feedback control in online advertising, in: Proceedings of the 2013 American Control Conference, 2013, pp. 6008–6013.
[21] N. Karlsson, J. Zhang, R. Luenberger, S.R. Strickland, Y.-S.C. Huang, S.M. Ziaemohseni, L. Yang, H.W. Uhlig, Systems and Methods for Displaying Digital Content and Advertisements over Electronic Networks, United States Patent Application 20130197994, USPTO, 2012.
[22] N. Karlsson, J. Zhang, S.R. Strickland, Systems and Methods for Sponsorship Fulfillment and Control, United States Patent Application 13/841,187, USPTO, 2013.
[23] N. Karlsson, J. Zhang, Y. Xu, Methods for Controlling an Advertising Campaign, United States Patent 7,835,937, USPTO, 2010.
[24] E. Kaufmann, O. Cappé, A. Garivier, On Bayesian upper confidence bounds for bandit problems, in: Fifteenth International Conference on Artificial Intelligence and Statistics (AISTAT), 2012.
[25] V. Krishna, Auction Theory, Academic Press, San Diego, California, 2002.
[26] T.L. Lai, H. Robbins, Asymptotically efficient adaptive allocation rules, Adv. Appl. Math. 6 (1985) 4–22.
[27] O.-A. Maillard, R. Munos, G. Stoltz, Finite-time analysis of multi-armed bandit problems with Kullback–Leibler divergences, in: Conference on Learning Theory (COLT), 2011.
[28] G. Owen, Game Theory, 3rd ed., Academic Press, San Diego, California, 1995.
[29] W.R. Thompson, On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, Biometrika 25 (3–4) (1933) 285–294.