

# Recurrent Neural Networks for Stochastic Control in Real-Time Bidding

Nicolas Grislain

Verizon Media

Paris, France

nicolas.grislain@verizonmedia.com

Nicolas Perrin

Verizon Media

Paris, France

nicolas.perrin@verizonmedia.com

Antoine Thabault

Verizon Media

Paris, France

antoine.thabault@verizonmedia.com

## ABSTRACT

Bidding in real-time auctions can be a difficult stochastic control task; especially in a very uncertain market and if underdelivery incurs strong penalties. Most current works and implementations focus on optimally delivering a campaign given a reasonable forecast of the market. Practical implementations have a feedback loop to adjust and be robust to forecasting errors, but no implementation, to the best of our knowledge, uses a model of market risk and actively anticipates market shifts.

Solving such stochastic control problems in practice is actually very challenging and an approximate solution based on a *Recurrent Neural Network* (RNN) architecture is both effective and practical for implementation in a production environment. The RNN bidder provisions everything it needs to avoid underdelivery. It also deliberately falls short of its goal when buying the missing impressions would cost more than the penalty for not reaching it.

## CCS CONCEPTS

• **Mathematics of computing** → **Probability and statistics**; • **Applied computing** → *Online auctions*.

## KEYWORDS

Ad-tech; Auctions; Real-Time Bidding; Recurrent Neural Network; RNN; RTB; Stochastic Control

## ACM Reference Format:

Nicolas Grislain, Nicolas Perrin, and Antoine Thabault. 2019. Recurrent Neural Networks for Stochastic Control in Real-Time Bidding. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19)*, August 4–8, 2019, Anchorage, AK, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3292500.3330749>

## 1 INTRODUCTION

Since its advent in 2009, *Real-Time Bidding* (RTB), *a.k.a. programmatic media buying*, has been growing very fast. In 2018, more than 80% of digital display ads are bought programmatically in the US [8].

RTB is allowing advertisers (buyers) to buy ad-spaces to digital publishers (sellers) at a very fine-grained level, down to the user

and the particular ad impression. The problem of using all the information available about each user exposed to some ad-placement to deliver a certain amount of impressions, clicks or viewable impressions, in an optimal way, is called *the optimal bidding problem*.

The optimal bidding problem may come in different flavors. It may be about maximizing a given *Key Performance Indicator* (KPI): impressions, clicks or views, given a certain budget, or about minimizing the cost of reaching some KPI goal. It is often formulated in a second price auction setup, but different setups, like first price auction or other exotic setups, are common on the market.

The problem of optimizing one campaign, competing with the market in second price auctions is considered. The campaign is aiming at a daily KPI goal, with a penalty for falling short of the goal. This restriction does not harm the generality of this work as most of it is generalizable to other sorts of goals and time spans. The campaign is small enough so that the impact of its delivery on the market is negligible.

Market data is eventually observable, which means it is possible to know, after some time (possibly hours), at what price a given impression would have been bought even if it was lost. This last assumption is valid in our practical setup where a company controls both an inventory and a bidder buying its own inventory on behalf of external advertisers, possibly in competition with third party bidders. This assumption can be relaxed though, at the cost of a more complex training process.

This paper puts a strong emphasis on the way market uncertainty is handled in a context where a fixed goal is to be achieved despite the stochastic nature of the market. Without uncertainty, our problem reduces to a relatively simple optimal control problem, adding randomness makes it an intractable stochastic control problem. We propose a characterization of the solution in terms of a *Partial Differential Equation* (PDE) and an approximate solution using a *Recurrent Neural Network* (RNN) representation.

The main contributions of this paper can be summarized as follows:

- (1) It formalizes in Section 3 the optimal bidding problem as a stochastic control problem, where market volume and prices are stochastic,
- (2) It solves numerically a simple case in section 4 and comments qualitatively the solutions,
- (3) It builds a practical RNN that approximate the theoretical solution in Section 5,
- (4) The RNN is trained and tested at scale on a major ad-exchange as described in Section 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

KDD '19, August 4–8, 2019, Anchorage, AK, USA

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6201-6/19/08...\$15.00

<https://doi.org/10.1145/3292500.3330749>

## 2 RELATED PAPERS

A description of the various challenges brought by the impression-level user-centric bidding compared to bulk, inventory-centric buying is done in [21]. [22, 23] gives a very broad overview of the optimal bidding problem.

[5] solves a bidding problem with multiple campaigns and from the perspective of the publisher using linear programming and duality. A similar question is solved in [3, 14]: the publisher wants to allocate impressions to campaigns in competition with third party RTB campaigns. [14] allows for underdelivery by introducing a *penalty* for underdelivery in its optimization program. [11] describes a solution to the bidding problem with budget constraints and partially observed exchange.

To account for market uncertainty, the optimal bidding problem is solved using a *Markov Decision Process* (MDP), constantly adapting to the new state of the campaign on the market. [10] proposes a heuristic in the field of yield management. [15–17] propose to use a *Proportional Integral* (PI) controller to control the bidding process and add some randomness to the bid to help exploration in a partially observed market and alleviate the *exploration-exploitation dilemma*. [4] uses dynamic programming to derive an optimal policy auction by auction, which makes the proposed methodology slightly impractical. [9] gives a very rigorous statement of the problem and solves it in cases where impressions are generated by homogeneous Poisson processes and market prices are *independent and identically distributed* (IID).

The general bidding problem with nonstationary stochastic volume and partially observed market is a complex *Reinforcement Learning* (RL) problem tackled in [20] using tools from the *deep reinforcement learning* literature. [20] uses, as is done in this paper, the common approach of bidding proportionally to the predicted KPI probability and solves a control problem over this proportionality factor every few minutes instead of optimizing for every impression. It makes the approach practical for real uses.

[20] finds the use of immediate reward misleading during the training, pushing to solutions neglecting the budget constraint. The proposed approach introduces budget constraints in the reward by simply adding a linear penalty. The bidder may explore the costly scenarios where it falls short of its goal and avoid them.

Also, the MDP trained in [20] uses a state engineered by the author, mainly: the current time step, the remaining budget, the budget consumption rate, the cost per mille at the last period, the last win-rate and the last reward. This choice is reasonable but the memory of the MDP is reduced to the remaining budget and what can be inferred from the last period. The approach proposed in this paper does not specify the state space and state transition, the *Recurrent Neural Network* (RNN) state is learned from the data. In particular *it can learn and encode the type of day or the type of shocks the market is undergoing* and reacts accordingly.

## 3 THE BIDDING PROBLEM UNDER UNCERTAINTY

In this section, the bidding problem is considered in the specific context of a bidder aiming at delivering campaigns in competition with the market, on media owned by itself. This does not harm the generality of the work, but explains the availability of sell-side

data for training and the kind of objective considered: number of impression at minimum cost. Without sell-side data, the training of the model exposed below would be made more complex by the censorship of market data for lost auctions.

### 3.1 Formal statement of the bidding problem

Let  $(\Omega, \mathcal{F}, \mathbb{P})$  be a probability space equipped with a filtration  $(\mathcal{F}_t)_{t \in \mathbb{R}^+}$ . A *bidder* is assumed to bid against the market representing all the competition. Let  $(\mathcal{H}_t)_{t \in \mathbb{R}^+}$  denote the sub-filtration encoding the restricted information accessible to the bidder.

All bidders receive ad requests modeled by the jumps of a Poisson process  $(I_t)_t$  with intensity  $\iota_t$ . For each impression opportunity  $i$ , happening at  $t_i$  the bidder receives  $x_i$ <sup>1</sup> the current features of the impression: timestamp, auction\_id, user\_id, placement\_id, url, device, os, browser or geoloc.

Based on  $x_i$  and, more generally, on all past history  $H_i \in \mathcal{H}_i$  a bid  $a_i$  is chosen  $a_i = a(H_i)$ . The bidder wins the auction whenever  $a_i > b_i$ , where  $b_i$  is the highest of the other bids in the market. Each impression has some value  $u_i$  to the bidder. When trying to buy clicks,  $u_i$  is 0 or 1 depending on the occurrence of a click.

The bidder is assumed to know its expected value  $v_i(H_i) = \mathbb{E}(u_i | H_i)$ , not to have a significant influence on the market as well as to have access to some distribution of  $b_i$  conditional on  $\mathcal{H}_i$ .

This paper characterizes an optimal bidding strategy  $a(H_i)$ .

### 3.2 The bidding problem over a short period of time

The bidder's spend follows the process:

$$dS_t = b_{I_t} \mathbf{1}_{a(H_t) > b_{I_t}} dI_t,$$

and the cumulative value follows:

$$dV_t = v(H_t) \mathbf{1}_{a(H_t) > b_{I_t}} dI_t.$$

Let us consider a short period of time  $\delta$  such that conditionally on  $\mathcal{H}_t$ ,  $\iota_t$  is predictable with average value  $\iota$  and  $x_i$ ,  $b_i$ ,  $u_i$  are *independent, identically distributed* (IID) over  $[t, t + \delta]$ . Let us consider that  $\delta$  and  $\iota$  are such that  $\delta \cdot \iota$  is large and its relative standard deviation small<sup>2</sup>:

$$\frac{1}{\sqrt{\delta \cdot \iota}} \ll 1.$$

Over a period of time  $[t, t + \delta]$ , the set of impression is noted  $I_t$  and the number of impression is almost deterministic  $I_t = \delta \cdot \iota_t$ . Because  $x_i$ ,  $b_i$ ,  $u_i$  are IID, each auction is brand new and the only relevant information for the bidder is  $x_i$ , that is  $a(H_i) = a(x_i)$ . In a Second Price Auction setup<sup>3</sup>, the spend is

$$S_I(a) = \sum_{i \in I} b_i \mathbf{1}_{a(x_i) > b_i}$$

and the value is

$$V_I(a) = \sum_{i \in I} u_i \mathbf{1}_{a(x_i) > b_i}.$$

<sup>1</sup>Variables  $X$  are indifferently noted  $X_{t_i}$  or  $X_i$ .

<sup>2</sup>In practice  $\delta$  would be in the order of magnitude of 100 *seconds* while  $\iota$  close to 1000 *events per second* so the relative error would be around 1%.

<sup>3</sup>See, e.g., [19] for an introduction to second price auctions.

Because the  $x_i$ ,  $b_i$  and  $u_i$  are IID and the values summed over a large number of impressions, everything becomes deterministic and reduces to

$$S_I(a) \approx I \cdot \mathbb{E}_x \left[ \int_0^{a(x)} b f(b; x) db \right]$$

and

$$V_I(a) \approx I \cdot \mathbb{E}_x [v(x) F(a(x); x)],$$

where  $f(b; x)$  is the *Probability Density Function* (PDF) of  $b$  conditional on  $x$  and  $F(b; x)$  is the *Cumulative Density Function* (CDF) associated.

The optimization program of the bidder can be written

$$\min_a C(a) = \min_a [S_I(a) + K \max(0, G - V_I(a))].$$

It can be read: *the bidder chooses a bidding strategy  $a$  such that its overall cost  $C$  is minimized, while its goal  $G$  (in number of impressions of clicks) is reached.* The cost is composed of the spend  $S_I$  incurred by the purchase of impressions, and a possible linear penalty  $K(G - V_I(a))$  paid if one falls short of the goal  $G$ <sup>4</sup>. This is not the most common formulation of the problem but it fits the practical need described above<sup>5</sup>. The *Karush-Kuhn-Tucker* (KKT) conditions give the following<sup>6</sup>:

$$\begin{aligned} a(x) f(a(x); x) &= \lambda v(x) f(a(x); x), & \forall x, \\ a(x) &= \lambda v(x), & \forall x, \end{aligned}$$

with  $\lambda \in [0, K]$ .

This means that the optimal strategy is to bid a value proportional to  $v(x)$ . If we restrict to the case where the bidder tries to buy a certain amount of impression at the best possible price, then the optimal strategy is to bid a constant bid. For the rest of this work, and without loss of generality, the bidder aims at buying a certain amount of impression, hence all  $u_i$  are 1,  $v(x) = 1$  and

$$a(x) = \lambda \in [0, K], \quad \forall x.$$

Most of these results do not hold for longer than  $\delta$  if the auctions are no longer IID or  $u_t$  non predictable. In practice, random external factors affect the total volume of impressions  $u_t$  in unpredictable ways. The market conditions is also prone to large shifts, e.g. when a very large campaign crowds out the other campaigns on a particular inventory.

As much as it is safe to assume everything is usually well predictable for the next few minutes, it is no longer the case for longer timescales.

<sup>4</sup>Note that the penalty does not have to be paid at the end of the short period of time, because the goal is additive, this short period of time can be combined with other periods of time as in the next section.

<sup>5</sup>A more common formulation would be to maximize the value, with a penalty for exceeding some budget:

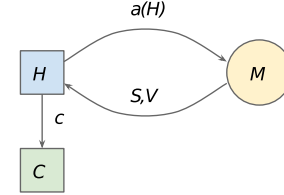
$$\max_a U(a) = \max_a V_I(a) - L \min(0, S_I(a) - B).$$

This is equivalent in the sense that very similar first order conditions are derived from both approaches.

<sup>6</sup>KKT covers the case when the goal is attained, the more general result would be derived from first order sub-gradient condition for convex minimization.

### 3.3 The bidding problem over a full day

Because the market is unpredictable, the bidder knows that no matter what he plans based on  $H_t \in \mathcal{H}_t$ , he has to adjust constantly to new available information to reach its daily goal. For this reason, the bidding strategy is now modeled as a *Markov Decision Process* (MDP) as in [4, 10, 15–17, 20], see Fig. 1.



**Figure 1:** In the MDP approach all the information available, e.g. all past spends  $S$  and volume purchased  $V$ , is embedded into the state  $H$ , the bid  $a(H)$  is built out of  $H$  and submitted to the market  $M$ . Market response ( $S$  and  $V$ ) is fed back to the bidder and constitutes the reward/cost  $c$ , adding up to the total cost  $C$ .

The full day is split in  $T$  periods of duration  $\delta$ . For each period, the bidder sets  $a$  based on  $t$ , the remaining goal  $G_t$  and a state  $H_t$  coding for everything he learnt from past experience that is relevant, and he observes the common distribution of all  $b_i$  whose PDF and CDF are noted  $f(b; H_t)$  and  $F(b; H_t)$ . Note that the bid distribution is fully determined by the current state  $H_t$ .

The expected spend given knowledge of the state  $H$  is

$$S(H, a) = I(H) \cdot \int_0^a b f(b; H) db$$

and the expected volume of impression is:

$$V(H, a) = I(H) \cdot \int_0^a f(b; H) db = I(H) \cdot F(a; H).$$

Let  $C_t(G, H)$  be the lowest cost achievable to deliver  $G$  impressions over  $[t, T]$  given knowledge  $H$ .

$$C_t(G, H) = \min_{(a_s)_{s \geq t}} \mathbb{E}_t \left[ \sum_{s \geq t} S(H, a_s) + K \max \left( 0, G - \sum_{s \geq t} V(H, a_s) \right) \right]$$

$C_t(G, H)$  is simply the sum of all spends  $S$  plus the penalty  $K$  times the *shortfall* given the optimal bidding strategy.

The optimal control is therefore fully characterized by the following Bellman equation:

$$\begin{cases} C_t(G, H) = \min_a \mathbb{E}_t [S(H, a) + C_{t+1}(G - V(H, a), \mathcal{T}_t(H))], \\ C_T(G, H) = K \max(0, G). \end{cases} \quad (1)$$

where  $\mathcal{T}_t$  is the transition function, taking the current state  $H_t$  and returning the next state  $H_{t+1} = \mathcal{T}_t(H_t, a_t)$ . Because  $a$  has no impact on the market, the transition function can be noted  $H_{t+1} = \mathcal{T}_t(H_t)$ .

The first order condition on  $a$  gives:

$$a_t(G, H) = \mathbb{E}_t \left[ \frac{\partial}{\partial G} C_{t+1}(G - V(H, a_t(G, H)), \mathcal{T}_t(H)) \right].$$

It can be noted that at each period  $t$ , the bidder optimizes for the current period, knowing it has to optimize for the remaining periods, up to  $T$ . Also, the optimal  $a$  is chosen equal to the marginal expected cost. When the goal is far from being reached,  $a$  will be high, else it will be low.

It can also be proved that  $a$  is in the interval  $[0, K]$ . It is clearly the case for  $t = T$ , but it is also the case for  $t$  as long as it is true for  $t + 1$ , because  $C_t$  as a function of  $G$  is a mixture of  $C_{t+1}$ , which suffices to show

$$a_t \in [0, K], \quad \forall t. \quad (2)$$

In the special case where  $\delta$  is small enough and  $H_t$  continuous, the problem can be usefully expressed in continuous time.

### 3.4 Solution in continuous time

Let us solve the optimal bidder problem in a continuous time setting with a simple Brownian motion model of available volume:

$$dH_t = \mu(t, H_t) dt + \sigma(t, H_t) dW_t, \quad (3)$$

where  $W_t$  is a one-dimensional Wiener process. The spend intensity at  $t$  is

$$S(H_t, a) = I(H_t) \cdot \int_0^a b f(b) db$$

and the volume intensity is

$$V(H_t, a) = I(H_t) \cdot \int_0^a f(b) db = I(H_t) \cdot F(a).$$

In this model, the available volume is stochastic but the bid distribution is constant.

The minimization problem writes

$$C(t, G, H) = \min_{(a_s)_{t \leq s \leq T}} \mathbb{E}_t \left[ \int_t^T S(H_s, a_s) ds + K \max \left( 0, G - \int_t^T V(H_s, a_s) ds \right) \right].$$

The Hamilton-Jacobi-Bellman (HJB) equation states that

$$\begin{aligned} \frac{\partial C}{\partial t} + \frac{\partial C}{\partial H} \mu(t, H_t) + \frac{1}{2} \frac{\partial^2 C}{\partial H^2} \sigma^2(t, H_t) \\ + \min_{a_t} \left[ S(H_t, a_t) - \frac{\partial C}{\partial G} V(H_t, a_t) \right] = 0, \end{aligned}$$

with the limit condition  $C(T, G, H) = K \max(0, G)$ .

At the minimum  $a_t$  verifies the first order condition

$$I(H_t) \cdot a_t f(a_t) = \frac{\partial C}{\partial G}(t, G_t, H_t) I(H_t) \cdot f(a_t),$$

which reduces to:

$$a_t = \frac{\partial C}{\partial G}(t, G_t, H_t).$$

The HJB equation can be solved in  $a(t, G, H)$ :

$$\frac{\partial a}{\partial t} + \frac{\partial a}{\partial H} \mu(t, H) + \frac{1}{2} \frac{\partial^2 a}{\partial H^2} \sigma^2(t, H) - I(H_t) \cdot F(a) \frac{\partial a}{\partial G} = 0,$$

with the limit conditions:

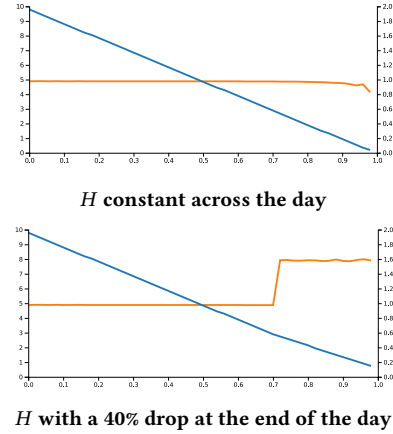
$$\begin{cases} a(T, G, H) = 0, & \text{if } G < 0, \\ a(T, G, H) \in [0, K], & \text{if } G = 0, \\ a(T, G, H) = K, & \text{if } G > 0. \end{cases}$$

## 4 A NUMERICAL RESOLUTION

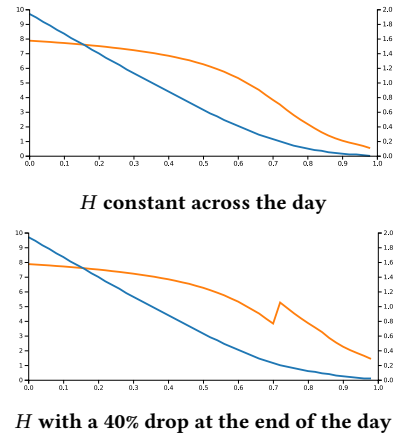
In the special case where  $\mu(t, H) = 0$ ,  $\sigma(t, H) = \sigma$ ,  $I(H) = H$  and  $F$  is the CDF of a gaussian distribution,  $a$  solves the partial differential equation

$$\frac{\partial a}{\partial t} + \frac{1}{2} \sigma^2 \frac{\partial^2 a}{\partial H^2} - H \cdot F(a) \frac{\partial a}{\partial G} = 0, \quad (4)$$

which can be solved numerically for various levels of uncertainty.



**Figure 2: The resolution of Eq. (4) with  $\sigma = 0$ . The blue curve shows the remaining goal to achieve across time. The orange curve shows the bid level. One can notice the sharp increase in bid after the shock and the goal shortfall.**



**Figure 3: The resolution of Eq. (4) with  $\sigma = 3$ . The bidder front-loads some of its delivery.**

The numerical solutions in Fig. 2 and Fig. 3 show that the introduction of uncertainty  $\sigma = 3$  in the market induces some provisioning behavior in the optimal strategy. This provisioning for risk is materialized by a decreasing bid with time whenever the risk does not materialize, which is the case in those simulations where  $H$  is constant except for a shock at the end of the day.

The MDP approach solves the bidding problem under uncertainty in a satisfactory way, but in the general case the relevant information in all the information available:  $H$ , is not obviously observable and using all  $H$  requires working in spaces too large to be practical.

The general case can be approximated. Such an approximation needs to be chosen in a functional space rich enough to capture the desired features. In Section 5, RNNs are tested to this end.

## 5 PRACTICAL MDP BIDDING

As demonstrated above, a robust bidding strategy should adjust its bidding behavior continuously based on the last information available, but also based on the fact it has to adjust in the future.

Building such a system is complex. Let us say a bidder records the history of all the bids  $b_i$ , spends  $S_i$ , purchased volumes  $V_i$  and any other relevant information  $x_i$  for each auction  $i$ . This sequence is noted

$$H_i = (X_0, X_1, \dots, X_{i-1}, X_i) \in \mathcal{X}^*,$$

where  $X_i = (b_i, S_i, V_i, x_i) \in \mathcal{X}$ .

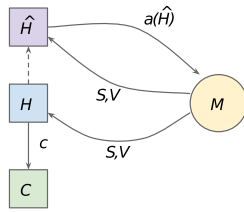
A bidding strategy should be a function  $a(t, G, H)$  of time, remaining goal and the finite sequences  $H \in \mathcal{X}^*$  to  $\mathbb{R}$ . Solving a minimization problem on such a space is largely intractable, even numerically, so we rely on some finite dimensional representation  $\hat{H} \in \mathbb{R}^n$  of  $H$ , enabling a fair approximation of the solution:

$$a(t, G, \hat{H}) \approx a(t, G, H).$$

The state  $\hat{H}_t$  is not updated for every auction, but instead at a regular pace. It is computed based on  $\hat{H}_{t-1}$ , the remaining time to deliver  $T - t$ , the remaining volume to reach  $G_t = G_{t-1} - V_{t-1}$  and the last spend  $S_{t-1}$ :

$$\hat{H}_t = \mathcal{T}(\hat{H}_{t-1}, V_{t-1}, S_{t-1}; \theta).$$

The transition function  $\mathcal{T}(\cdot; \theta)$  is trained to minimize the cost of the campaign (cf. Fig. 4).



**Figure 4: In the practical MDP approach all the information available, e.g. all past spends  $S$  and volume purchased  $V$  are used to update a finite dimensional *proxy* state  $\hat{H}$ . The bid  $a(\hat{H})$  is submitted to the market  $M$ . The state transition function (the way  $\hat{H}$  is updated) is trained from data to minimize total cost  $C$ .**

In the next two sections, two different practical implementations of a bid controller to provide an approximation of the solution are presented: the *Proportional Integral controller* in Section 5.1, and the *Recurrent Neural Network controller* in Section 5.2.

### 5.1 The Proportional Integral controller

The Proportional Integral (PI) controller<sup>7</sup> is widely used in various industries [7]. [15, 22] propose to apply it to the bidding problem<sup>8</sup>.

The interaction between the bidding agent and the market can be modeled as a feedback system composed of a *feedback controller* and a *block M* representing the RTB market. The system receives as input a *reference signal*  $\hat{V}_t = \hat{V}(t, G_t; \theta_V)$ : a target volume for the next time period.

From the feedback  $V_t$  received from the market, the controller computes a *control error*  $e_t = \hat{V}_t - V_t$ . Based on it, the controller maintains a state and uses it to generate a new control variable (or action of bidding at a specific bid level)  $a_t$

$$a_t = \theta_P e_t + \theta_I \sum_{s=0}^t e_s,$$

where  $\theta_P, \theta_I$  are called the *proportional* and *integral gains*.

In the PI setup, the state and its transition function  $\mathcal{T}(\cdot; \theta)$ , where  $\theta = (\theta_V, \theta_P, \theta_I)$ , can be expressed

$$\hat{H}_t = \begin{pmatrix} 0, 0, 0 \\ 0, 1, 0 \\ 0, \theta_I, 0 \end{pmatrix} \hat{H}_{t-1} + \begin{pmatrix} 1 \\ 1 \\ \theta_P + \theta_I \end{pmatrix} (\hat{V}(t-1, G_{t-1}; \theta_V) - V_{t-1})$$

and

$$a(t, G_t, \hat{H}_t) = \left\langle \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \hat{H}_t \right\rangle.$$

Training the PI controller is done in two steps: a reference forecasted volume process  $\hat{V}$  is defined and trained, and then the gains are tuned using Stochastic Gradient Descent.

Although simple and robust, this approach comes with some flaws.

- (1) It depends on a separate forecaster  $\hat{V}$ .
- (2) It is designed to target a current value, not to optimize a lifetime cost function; this is mitigated by the fact the parameters are tuned against the lifetime cost.
- (3) The uncertainty about the market is not modeled, it is barely taken into account through the cost function, but no component of the state  $\hat{H}$  really reflects anything about risk.
- (4) The important gap between the small number of parameters of the PI model and the large amount of data available suggests probable underfitting. Capacity can be added to the model by allowing adaptive gains, setting thresholds and special cases, but those are merely local patches.

To overcome these flaws, we introduce in the next section a new approach leveraging a Recurrent Neural Network to approximate the bidding problem solution. A PI controller is used as benchmark to the RNN approach.

<sup>7</sup>See [2, Chapter 10] for an introduction to the PI control.

<sup>8</sup>The motivation for choosing a PI controller rather than a Proportional Integral Derivative (PID) controller is twofold. First, the presence of noise can lead to undesired drastic changes in the controller output due to derivative action. Second, the models used in the target production infrastructure at the time of the writing are PI controllers.

## 5.2 The RNN controller

The Recurrent Neural Network (RNN)<sup>9</sup> controller unit used in all the presented experiments is a Gated Recurrent Unit (GRU, see [6])<sup>10</sup>, with

**input:** a vector  $(T - t, G_t, V_t, S_t)$ ,

**state:** a vector  $\hat{H}$  with dimension  $16^{11}$ ,

**activation:** a hyperbolic tangent function rescaled for the first component of the state  $\hat{H}$  to be between 0 and the penalty level  $K^{12}$ ,

and where the bid level is given by the first component of the state  $\hat{H}$  of the GRU layer:

$$a(t, G_t, \hat{H}_t) = \left\langle \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \hat{H}_t \right\rangle.$$

Through the recurrent connections the model can retain information about the past in its state, enabling it to discover temporal correlations between events and its interactions with the environment even when these are far away from each other in the data. Using a RNN allows to take advantage of a much richer family of functions to learn an approximate solution to the bidding problem.

## 6 EXPERIMENTS

### 6.1 Practical setting

In practice, the massive number of auctions occurring simultaneously makes unrealistic the resolution of the optimal control (1) for each auction and campaign. Fortunately, taking periodic control decisions (*e.g.* every 5 minutes) on aggregated feedback is sufficient. It is thus possible to handle a very large number of campaigns, with the following steps:

- at the beginning of each period, choose a level for the control variable  $a$ ,
- get an aggregated feedback (realized volume and spend) from the previous period in response to the level  $a$ .

This kind of architecture introduces discontinuity in the response of the controlled advertising system. Yet, the problem can be turned into a continuous control problem [16, 17].

Furthermore, response curves exhibit discontinuities (Fig. 5, left plot). Those can be smoothed out (Fig. 5, right plot) by drawing bid prices according to some distribution (*e.g.* log-normal, Gamma, *etc.*) around the control variable  $a$  (such as proposed in [15, 17]) instead of bidding a constant bid level during the time period.

This leads to the following loop:

- at the beginning of each period, choose a level for the control variable  $a$ ,
- for each auction occurring in the period, draw a bid price according a distribution based on  $a$ .

<sup>9</sup>See [12, Chapter 10].

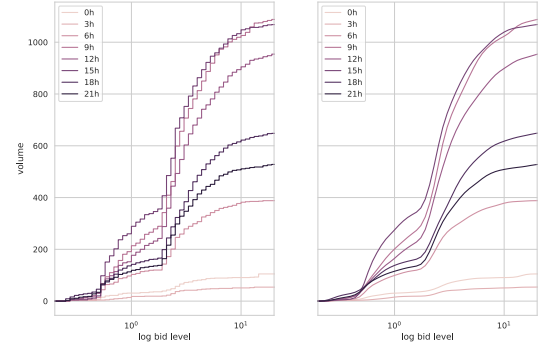
<sup>10</sup>Long Short-Term Memory (LSTM) [13] was also experimented with but the results were similar to the GRU ones.

<sup>11</sup>Simple trials were also conducted to assess the interest of using more neurons in recurrent units in the RNN architecture, be it wider or deeper. No significant gain was found, and a detailed assessment lies beyond the scope of this paper.

<sup>12</sup> $K$  is the highest possible bid in an optimal strategy, *cf.* Eq. (2).

- get an aggregated feedback (realized volume and spend) from the previous period.

Off-policy training of the models is achieved everything else being equal, *i.e.* by replaying auctions under the assumption that the third-party bids are not impacted.



**Figure 5: An example of bid level to volume mapping evolution through the day. Left: no noise (Dirac bids). Right: volume obtained by Gamma-distributed bids around each bid level.**

Note that contrary to the general setup of the bidding problem that suffers from censorship due to the ad auction selection [22], one can alleviate this particular issue in this practical setup since publisher data is available. In the absence of uncensored data, the bid randomization would further help, by realizing some of the exploration effort in the *explore vs exploit* dilemma introduced by bid-dependent censorship.

### 6.2 Data used

Two types of datasets are used in the numerical experiments presented:

- (1) Simulated synthetic data sets, generated by applying various transformations (shocks or random walk) throughout time to a base linear volume curve, that helps in appreciating salient features of the RNN models,
- (2) Production RTB data, constituted by 5-minute snapshots of the actual price-volume mapping for all display ad placements with significant daily volume<sup>13</sup> from large publishers on one of the leading *Supply Side Platform* (SSP) and Ad Exchange globally. The ad placements here can be seen as proxys of segments targeted by campaigns. In production, the RNN would have to be trained on currently running campaigns.

A simulated case study is conducted using generated datasets where the available volume is a linear function of the bid<sup>14</sup> and its

<sup>13</sup>Restricted to the placements with a minimum of 1000 daily impressions.

<sup>14</sup>The initial slope is chosen so that bidding \$CPM 5 at every period would deliver 100 impressions in the absence of noise.



process follows a random walk as in Eq. (3) with no drift term and various levels of noise. In other words, the slope of the bid to volume relationship changes through time according to the uncertainty of the volume process. In one experiment, we additionally introduce a permanent shock in the evaluation scenario to study the impulse response of the model. The type of shock used is a permanent change of the slope after a given date.

The production dataset is created using logs from actual RTB auctions run on around 1000 ad placements over 8 days, containing about 115M won impressions. All these impressions are used to build winning bid distributions for 5-minutes periods over a full day ( $T = 288$ ) of each ad placement. The winning bid distributions are discretized on a CPM bid scale with 100 exponential increments between 0.01 and 100.

For offline training and evaluation purposes on production data, a bidding problem instance is comprised of a random draw of an actual bid-volume mapping process and of a random volume goal, uniformly drawn between 10 and 1000. The controller therefore is exposed to scenarios with not enough of volume to meet the target given the penalty level, as well as scenarios where enough volume was available.

The production data is split into non-overlapping training, validation and evaluation datasets using different days. The training set of models on production data contains 1 million different bidding problems and evaluation is performed on a set of around 110K bidding problems of increasing difficulty. For the simulated case study, given the simplified setting, training is stopped after learning from 20K bidding problems.

### 6.3 Training and evaluation

The implementation<sup>15</sup> of both the benchmark (PI controller) and the RNN controller is done in TensorFlow [1]. The input data instances are randomly shuffled and processed by batches of size 100.

The aim is to minimize the total cost of a campaign, so the training loss  $C$  is composed of the sum of the spend and the penalty terms over a full day:

$$C := \sum_{s=0}^T S_s + K \max \left( 0, G - \sum_{s=0}^T V_s \right)$$

where the spend  $S_t$  and volume won  $V_t$  are computed from the bid level  $a_t$  of the MDP bidding controller by simulating the feedback using the input bid distribution at each time step and propagating the state over the full sequence of time periods.

Models are trained using *Stochastic Gradient Descent* (SGD) with an inverse-time decay of the learning rate<sup>16</sup>. To help alleviate possible exploding gradients issues, gradient clipping is used as described in [18]. Other more sophisticated optimization methods have been tried without significant impact on the results.

Cross-validation is performed regularly during the training on a fixed set of validation bidding problems and the optimal model parameters are picked as the best evaluation seen on the validation set during the training. In practice, no model presented any

overfitting issue as performance results generalized well to unseen data.

### 6.4 Numerical results

**6.4.1 Simulated case study.** A first experimental setup on simulated data goes back to the simplified setting from Section 4. Fig. 6 displays the bid level, volume and spend during the day, along with the final delivery cost decomposed into the final spend and penalty for five different RNN models. Each RNN model is trained on a simulated dataset for which the volume process follows Eq. (3) with a different level of noise (and no drift term). Each column evaluates the same model on six cases without noise, for which a permanent shock in the available volume happens for all dates  $t \geq 65$ . A shock factor of  $\delta$  means that after the shock the available volume given the same bid level is divided by  $\delta$ .

Note that the RNN model is able to learn a good approximation of the optimal strategy derived theoretically. The RNN controller exhibits the same behavior as the one evidenced in Fig. 2 and Fig. 3 by exact resolutions of the control problem. Indeed, the bid strategy is constant when noise is absent during the training. Hence the volume acquired and spend linearly grow through time in the scenario without any shock in the available volume. In scenarios with permanent shock factors, this model carries on bidding at the same level, thus falling short of the volume target of 100 impressions. As the final volume shortfall grows with the shock factor, the bigger the shock occurs, the bigger the end penalty is.

One can observe that the more noise a RNN controller model was trained with, the higher it starts bidding and the easier it absorbs shocks in the available volume. The models trained with more uncertainty in the available volume tend to provision for the shortfall risk by bidding higher and earlier than the optimal level in the absence of noise. To the extreme, the optimal solution is to buy as fast as one can (without bidding higher than the penalty). This obviously entails a higher spend when the risk does not materialize. However, the more risk was anticipated by a model through its training, the lower the final spend and penalty are when risk do realize, demonstrated here by increasing level of permanent shocks.

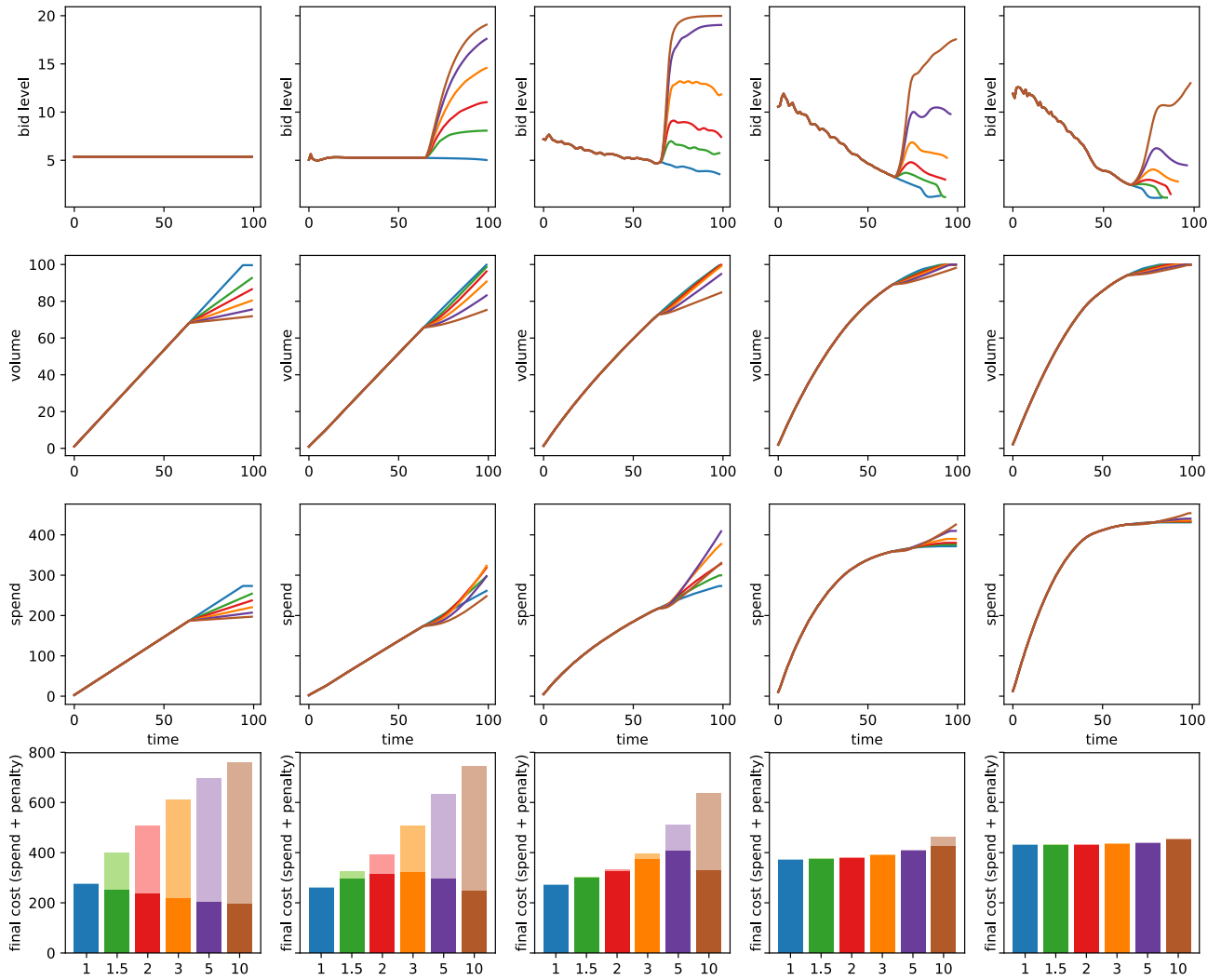
In a second experimental setup on simulated data, two RNN models were trained with either a low or high<sup>17</sup> level of noise and evaluated under both a low or high noise scenario. Main results are shown in Table 1. The experiment shows as expected that for both models the cost of delivery and the bidding problem difficulty (probability of shortfall) are higher in the evaluation scenario with high noise. However, given an evaluation scenario, i.e. an amount of noise that realizes, the delivery cost is lower for the model that has been trained on data with the same amount of uncertainty.

**6.4.2 Training on production data.** Performance results on actual market data are available in Table 2. As a benchmark, a PI controller is tuned to follow a reference pacing curve fitted on the training data. Indeed, a good approximation of the internet traffic intraday seasonality can be obtained using a model with only two harmonics [15]. The PI and RNN models are trained in the same

<sup>15</sup>Code is available at: [https://github.com/alephd/rnn\\_controller](https://github.com/alephd/rnn_controller)

<sup>16</sup>The learning rate schedule is the following:  $\alpha(n) = \frac{\alpha_0}{1+\eta \lfloor \frac{n}{N} \rfloor}$  with initial learning rate  $\alpha_0 = 0.1$ , decay rate  $\eta = 0.5$ , and decay steps  $N = 400$ .

<sup>17</sup>The high noise scenario is calibrated to be consistent with the average level of daily noise observed on our real data, its interquartile range being [7., 18.]. So the low noise scenario does represent a situation where risk is deeply underestimated.



**Figure 6: Bid strategy of RNN controller models trained with an increasing level of noise, and their reaction to permanent volume shock scenarios.** Each column shows the bid level, volume and spend throughout time as well as the final cost (sum of spend and penalty) of the same RNN model trained on simulated data with an increasing level of noise. Columns 1 to 5 respectively correspond to standard deviations of 0, 0.2, 1, 5, and 10. during the training. The models are here evaluated on deterministic data without noise. In each column, the six lines or bars represent various shock factors impacting the available volumes for all dates  $t \geq 65$ . A shock factor of  $x$  means that after the shock the available volume given the same bid level is divided by  $x$ . The bottom row decomposes the final cost incurred into the spend for buying the final impressions (bottom dark-colored bars) and the penalty that may be received if the volume target is not reached (light-colored stacked bars).

Mean final cost (std. dev.), shortfall probability		Evaluation scenario	
		$\sigma = 0.1$	$\sigma = 10$
Learning	$\sigma = 0.1$	\$271 (\$13), 5%	\$915 (\$516), 82%
scenario	$\sigma = 10$	\$431 (\$9), 0%	\$764 (\$405), 57%

**Table 1: Simulation results of RNN models trained and evaluated under high or low noise scenarios.**

conditions (same training, validation and test datasets) to ensure a fair comparison.

The Table 2 details the average total cost of delivering campaigns of increasing daily volume goals for both the PI model and the RNN model. Overall, the RNN model is able to reduce delivery cost by about 20% compared to the PI model. As the volume target increases, so does the bidding problem difficulty as the total available volume under the penalty level is constraining the bid strategy for a larger share of the dataset. Eventually, for very large volume goals the optimal strategy is to bid the penalty level, capturing all the volume



Goal (imps)	Delivery cost (\$CPM)		ratio RNN/PI
	PI	RNN	
100	1.02	0.82	0.80
500	1.28	1.02	0.79
1000	1.60	1.27	0.80
1500	2.06	1.76	0.85

**Table 2: Model performance comparison on actual market data. Both models are trained and tested in similar conditions (offline simulations with the same training, validation and test datasets). The table shows a significant reduction of the delivery costs (\$CPM) in favor of the RNN for different values of the considered KPI (number of impressions to deliver).**

below this level and paying the penalty for each missed impression. Thus lower performance improvements are expected for the larger goals, relative to the size of the targeting.

## 7 CONCLUSION AND FUTURE WORKS

The RNN controller model proposed provides both an effective and practical method to solve the optimal bidding problem. It has the advantage not to rely on manually engineered features to represent knowledge about the current state or history that could be leveraged in a bidding strategy, but instead infers it from the data. For instance, a more advanced, adaptative, PI controller could be employed to tackle the control problem, e.g. by using splines modeling the price-volume mapping to efficiently store the response gain at various price points. However, such a model would still lack useful elements from the very complex state space it evolves in, mainly because it overlooks the impact that uncertainty about future market volume and bid landscape has on the optimal strategy.

Numerical experiments demonstrate that the proposed approach is able to improve significantly on existing bidding controllers, while being trainable and usable at production scale. The approximation of the state and space transition provided by the RNN leads to a solution that captures a key aspect of the solution, namely provisioning against the risk of underdelivery.

This work could be extended in many ways:

- The observability of all bids including those of lost auctions is convenient in the case of our work. This assumption could nevertheless be relaxed. The reconstruction of bid distributions for training would probably be more complex and the noise added to the bid would need to be used as an exploration device.
- In practice, setting multiple goals would be an interesting feature to add, e.g. buying impressions with some guarantee of viewability. The equations would be marginally changed.
- The first price and exotic auction cases add a significant amount of complexity to this approach, however those questions would be resolved at the impression scale, while the macroscopic (5 min) scale control problem would probably hold in a similar way.
- Giving the RNN some more feedback, based on the noise injected in the bid could probably help.

More generally, this paper shows how RNNs can be applied to complex control problem with success.

## REFERENCES

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. <https://www.tensorflow.org/> Software available from tensorflow.org.
- [2] KJ Åström and Richard Murray. 2008. *Feedback Systems: An Introduction for Scientists and Engineers* (1 ed.). Princeton University Press.
- [3] Santiago R Balseiro, Jon Feldman, Vahab Mirrokni, and S Muthukrishnan. 2014. Yield optimization of display advertising with ad exchange. *Management Science* 60, 12 (2014), 2886–2907.
- [4] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. 2017. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 661–670.
- [5] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. 2011. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1307–1315.
- [6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).
- [7] Lane Desborough and Randy Miller. 2002. Increasing Customer Value of Industrial Control Performance Monitoring - Honeywell's Experience. *AIChE Symposium Series* 98 (01 2002).
- [8] eMarketer. 2018. More than 80% of Digital Display Ads Will Be Bought Programmatically in 2018.
- [9] Joaquin Fernandez-Tapia, Olivier Guéant, and Jean-Michel Lasry. 2016. Optimal real-time bidding strategies. *Applied Mathematics Research eXpress* 2017, 1 (2016), 142–183.
- [10] Guillermo Gallego and Garrett Van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science* 40, 8 (1994), 999–1020.
- [11] Arpita Ghosh, Benjamin IP Rubinstein, Sergei Vassilvitskii, and Martin Zinkevich. 2009. Adaptive bidding for display advertising. In *Proceedings of the 18th international conference on World wide web*. ACM, 251–260.
- [12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT press.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [14] Grégoire Jauvin and Nicolas Grislain. 2018. Optimal Allocation of Real-Time-Bidding and Direct Campaigns. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 416–424.
- [15] Niklas Karlsson. 2014. Adaptive control using Heisenberg bidding. In *2014 American Control Conference*. IEEE, 1304–1309.
- [16] Niklas Karlsson. 2016. Control problems in online advertising and benefits of randomized bidding strategies. *European Journal of Control* 30 (2016), 31–49.
- [17] Niklas Karlsson. 2017. Plant gain estimation in online advertising processes. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2182–2187.
- [18] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR, abs/1211.5063* 2 (2012).
- [19] Tim Roughgarden. 2016. *Twenty lectures on algorithmic game theory*. Cambridge University Press.
- [20] Di Wu, Xiujun Chen, Xun Yang, Hao Wang, Qing Tan, Xiaoxun Zhang, Jian Xu, and Kun Gai. 2018. Budget constrained bidding by model-free reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1443–1451.
- [21] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. ACM, 3.
- [22] Weinan Zhang. 2016. *Optimal Real-Time Bidding for Display Advertising*. Ph.D. Dissertation. University College London.
- [23] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1077–1086.