Supplement

1 Mathematical Formulation of Bidding Strategy and Online Bidding Problem

We focus on designing a bid optimization strategy for RTB auctions. When facing a bid request together with an impression, a DSP evaluates specific values of the impression, and then an impression-level bid optimization strategy maps the values, advertisers' requirements, and remnant budget to a real-time bid price.

1.1 Technical terms and notations

First, some technical terms and notations are explained; refer to [Google.com(2022), Zhang(2016)] for more details.

(a) RTB-related terms and notations.

- **Impression**, an opportunity to show an ad in front of users.
- **Bid request**, a call for pricing an impression together with a message containing the impression feature.
- **Ad campaign**, advertisers promote their products by setting up an ad campaign containing impression features of interests, KPIs, CPA constraints, etc.
- **Auction scale**, denoted by $T \in \mathbb{Z}_+$. The estimated total number of bid requests an ad campaign receives within a specific interval, usually one-day.
- **Click-through rate, CTR**, denoted by $p^{\text{CTR}} \in [0, 1]$. It is the estimated probability that users click through an ad when they see it.
- **Conversion rate, CVR**, denoted by $p^{\text{CVR}} \in [0, 1]$. It is the estimated probability that users take predefined action after clicking through an ad.

(b) Impression-related terms and notations.

- **Impression feature**, denoted by a high dimensional vector, $\mathbf{x} \in \mathcal{X}$. Various features can describe an impression, such as cookie information, time, location, etc. The feature $\mathbf{x}$ can be regarded as an integrable random variable with an integrable probability density function (PDF) $p_{\mathbf{x}}$, which follows the independent identically distribution (i.i.d) assumption [Zhang(2016), Yuan et al.(2013)]. The latter assumption helps us use a simple generic stochastic model to design bid strategies and pay more attention to bid optimization.
- **Impression value**, denoted by $v \geq 0$. It is the estimated KPI of an impression if an advertiser wins the auction. It is provided by a feature mapping module that maps $\mathbf{x}$ to v [Zhang et al.(2021)], that is, $v = v(\mathbf{x})$.
- **Impression action value**,denoted by $v_{a,j} \geq 0$. It is the estimated amount of action(s) an impression takes if an advertiser wins the auction [Zhang(2016)]. There may exist k interested actions simultaneously that the subscript $j \in \{1, \ldots, k\}$ is used to distinguish them. It is also provided by a feature mapping module, that is, $v_{a,j} = v_{a,j}(\mathbf{x})$. It is different from v because of it is usually used to measure some intermittent performance, such as subscription, add to cart, and so on [Yuan et al.(2013),Han et al.(2017)].

(c) Bid optimization problem-related terms and notations.

- **Key performance indicator, KPI**. It is a quantitative measurement of advertising performance [Yuan et al.(2013)] that advertisers are interested in.
- **Bid price**, denoted by $b \geq 0$. The cost that an advertiser wants to pay for an impression being auctioned.
- **Winning price**, denoted by $w \geq 0$. For an impression, winning price is the lowest bid price to win its auction and $w = w(\mathbf{x})$ [Zhang(2016)]. For an auction, the relationship among bid price, winning price, and corresponding auction result is,

$$\begin{cases} \text{Advertiser wins}, & b \geq w \\ \text{Advertiser loses}, & b < w \end{cases} .$$

- **Cost**, denoted by $c \geq 0$. It is the cost for the impression the advertisers win, which relates to $\mathbf{x}$, the auction mechanism and the billing method of DSP [Zhang(2016),Zhang and Xu(2014),Yuan et al.(2013)]. In this paper, we use the second-price auction, a mechanism in which each bidder provably tends to price on their desires in accordance with the truthful value of $\mathbf{x}$ [Vickrey(1961)]. Besides, the billing method is another factor affecting the cost [Google.com(2022)]. In this paper, we assume the billing method is click-based, that is, For example, in click ads, advertisers only pay for clicked-through ads [Han et al.(2017)]; while in display ads, billing has nothing to do with whether an ad has been clicked or not [Yang et al.(2019)]. Thus, we define $c = c(\mathbf{x}, *)$, where $*$ means the billing method.

- **Budget constraint**, denoted by $B \geq 0$. It is the total cost the advertiser can use for an ad campaign during its lifetime.
- **CPA constraint**, denoted by $C \geq 0$. It is the maximum average cost the advertiser can pay for a specified action observed on the delivered impression and usually is one-day-based. For example, CPC constraint means the maximum average cost the advertiser pays when a user clicks on the delivered ad of the won impression.

1.2 Problem of Interests

In the bid optimization problem, we aim to find a strategy π that, by mapping the impression feature $\mathbf{x}$, and the remnant budget ΔB (equivalently, the consumed budget) to a bid price b, optimizes the expected KPI and holds CPA constraints. However, $\mathbf{x}$ is a constructed, high-dimensional vector, e.g., the methodology proposed in [Zhang and Xu(2014)], and objectives and constraints vary for advertisers. These complicate the direct use of $\mathbf{x}$ and the corresponding bidding strategy generalization. Therefore, there exists a supplied module called feature mapping, which maps x to specific predefined values [Yuan et al.(2014),Zhang et al.(2021),Lyu et al.(2022)], i.e., the impression value $v(\mathbf{x})$ and the impression action value $v_a(\mathbf{x})$. Feature mapping is another vital problem in RTB and due to the limited space, we assumed it is prior to this work. Then, we write $b_t = \pi_t(B_t, \mathbf{x}_t)$, where B_t is the consumed budget at time t, and model the bid optimization as

$$
\begin{aligned}
&\underset{\pi_t,\ t\in\{1,\ldots,T\}}{\arg\max}\ \mathbb{E}\left[\sum_{t=1}^{T} v(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))\right] \\
\text{s.t. } &\sum_{t=1}^{T} c(\mathbf{x}_t, *)\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t)) \leq B \\
&\frac{\sum_{t=1}^{T} \mathbb{E}\left[c(\mathbf{x}_t, *)\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))\right]}{\sum_{t=1}^{T} \mathbb{E}\left[v_{a,j}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))\right]} \leq C_j,
\end{aligned}
\tag{1}
$$

where $j = 1, \ldots, k$, indicator function $\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))$ means the t-th auction result, if $\pi_t(B_t, \mathbf{x}_t) \geq w(\mathbf{x}_t)$, then $\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t)) = 1$; otherwise, it equals to 0. The winning probability $g(b_t, \mathbf{x}_t)$ is the expectation of $\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))$ with respect to x_t.The objective in the above is to find $\pi_t, t \in \{1, \ldots, T\}$ maximizing the total amount of impression value v. Nevertheless, $\mathbf{x}$ is a random variable with $p_{\mathbf{x}}$, the objective is taken in the expected sense. The first constraint means the consumed budget is limited by a predefined B; we should bid carefully for each auction because we may miss desired impressions when the budget depleted. The second constraint measures the cost per actions determined by advertisers, such as cost per click, cost per conversion. etc.

1.2.1 A real-world example

As an example, we present a real-world business service [Han et al.(2017)] provided by *Taobao.com* to illustrate the functional optimization problem (1). Its main settings contain:

- **Maximizing objective**, $v(\mathbf{x}) = p^{\text{CTR}}(\mathbf{x})p^{\text{CVR}}(\mathbf{x})$, is the expected total number of transactions.
- **Constraints**, advertisers set a total budget constraint B, and there is only a cost per click constraint C for a day.
- **Billing method**, advertisers only pay for those clicked-though ads they won, abbreviated as 'click'. In this case, the cost is $c(\mathbf{x}, *) = c(\mathbf{x}, \text{'click'})$.
- **Others**. The only considered impression action value $v_a(\mathbf{x}) = p^{\text{CTR}}(\mathbf{x})$. The second-price auction is used; cost equals the winning price, that is, the second-highest bid price.

To sum up, its bid optimization problem is

$$\begin{aligned}
&\underset{\pi_t, t\in\{1,\ldots,T\}}{\arg\max} \ \mathbb{E}\left[\sum_{t=1}^{T} p^{\text{CTR}}(\mathbf{x}_t)p^{\text{CVR}}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))\right] \\
&\text{s.t.} \ \sum_{t=1}^{T} c(\mathbf{x}_t, \text{'click'})\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t)) \leq B \\
&\frac{\sum_{t=1}^{T} \mathbb{E}\left[c(\mathbf{x}_t, \text{'click'})\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))\right]}{\sum_{t=1}^{T} \mathbb{E}\left[p^{\text{CTR}}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))\right]} \leq C.
\end{aligned} \tag{2}$$

1.3 Markov Decision Process (MDP) Modelling

The problem (1) is a typical sequential decision making problem. A remarkable class of well-studied sequential decision-making problems is the Markov decision process (MDP) or reinforcement learning, where MDPs are applied to more complex tasks and solved in a data-driven manner. In this part, we will formulate (1) into an MDP problem. To do so, we need to change the formulation of (1) by redefining the objective with its constraints as punishment terms. The reason for doing so is that whether a constraint of (1) is violated or not cannot be tested from real-time accessed data. In this part, we consider a finite time-horizon Markov decision process $\{\mathcal{S}, \mathcal{B}, p, r, T\}$:

- **State set** $\mathcal{S}$. For each start of the t-th auction, the consumed budget B_t and the current impression feature $\mathbf{x}_t$ are the state $s_t = [B_t \ \mathbf{x}_t^\top]^\top$.
- **Bid price set** $\mathcal{B}$. For the t-th auction, bid price is b_t.

- **State transition probability**, $p = \mathbb{P}(s_{t+1}|s_t, b_t)$. For a state B_t and bidding b_t, the transition of B_{t+1} is:

$$\begin{cases} B_{t+1} = B_t + c(\mathbf{x}_t, *)\mathbf{1}(w(\mathbf{x}_t), b_t), & \text{if } B_t < B \\ B_{t+1} = B_t, & \text{if } B_t \geq B \end{cases}.$$

  The sequence of $\mathbf{x}_t$ is an i.i.d. random process (see Section 1.1), that is, the transition of $\mathbf{x}_{t+1}$ is not affected by action b_t and is fully described by $p_{\mathbf{x}}$. The above two transition rules together characterize the state transition probability.

- **Reward**, $r = r(s_t, b_t)$, which is slightly complicated because of the punishment of constraints,

$$r_t = v(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), b_t) - \sum_{j=1}^{k} \lambda_j c(\mathbf{x}_t, *)\mathbf{1}(w(\mathbf{x}_t), b_t) - \lambda_j C_j v_{a,j}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), b_t)$$

  where $\lambda_j > 0, j \in \{1, \ldots, k\}$ is the Lagrange multiplier that remakes the objective with the second constraint in (1) as punishment terms.

- **Time horizon**. The auction scale $T \in \mathbb{Z}_+$ is the time horizon.

Furthermore, we assume that functions v, v_a, g and $p_{\mathbf{x}}$ are known by the DSP prior to solving the bid optimization problem. In a real DSP, these functions mainly come from feature mapping and value estimation modules, such as CTR prediction [Guo et al.(2017), Zhou et al.(2018)], CVR prediction [Wen et al.(2021), Wen et al.(2019)], winning price and winning rate prediction [Wu et al.(2018), Li and Guan(2014), Shih et al.(2020), Zhu et al.(2017)], landscape prediction [Wang et al.(2016)], feature construction [Zhang and Xu(2014)], and so on. They are important in RTB, however their prediction is beyond the scope of this paper. The bid optimization under the MDP model is to solve:

$$\mathop{\arg\max}_{\pi_t, t=1,\ldots,T} \sum_{t=1}^{T} \mathbb{E}\left[r_t\right] \tag{3}$$

Note that the budget constraint has been encoded in the system dynamics, which does not explicitly appear in the above problem. Besides, in most previous optimization-based studies [Han et al.(2017), Karlsson(2020), Tunuguntla and Hoban(2021), Yang et al.(2019), Zhang(2016)], researchers do not consider the influence of the dynamic consumed budget and thus solve a constraint optimization problem only with a fixed predefined total budget and CPA constraints. These methods can hold optimality but are fragile in the face real uncertainty and disturbance. On the other hand, in some machine-learning-based studies [Cai et al.(2017), Grislain et al.(2019), He et al.(2021), Jin et al.(2018), Ren et al.(2017)], budget dynamics have been considered in MDP modelling. Nevertheless, they need enormous resources and data to train the networks. Besides, the question about the optimality still exists. Compared to these two research approaches, we focus on solving the bid optimization problem using the MDP modelling together with optimization methods.

2 Rollout Mechanism and Structure of Bidding Functions

Though we can use the classical dynamic programming (DP) methodology to solve (3), it is still challenging because:

(a) The MDP problem (3) is not completely equivalent to (1).

(b) The impression feature $\mathbf{x_t}$ is usually of high dimensions and sparse. It is inefficient to work out and then store the bidding strategy mapping the feature, among others, to the bid price.

(c) On Taobao, half of the ad campaign received more than 10^4 bid requests a day; in other words, they bid every 10 seconds on average. Moreover, if we consider the distribution of impressions within a day, the frequency will be higher. High decision frequency and large-scale data place high demand on infrastructure and algorithms.

(d) The rewards and transition of the consumed budget are time-delayed [Yuan et al.(2013)]. As the example in Section 1.2.1, an advertiser wins auctions and pushes ads in front of users. Whether the latter trades or not is usually not immediately known.

The above is in two parts: methodology and implementation. The methodology part is discussed in this section, which requires us to fill the model gaps and design an efficient solution. The implementation trade-off is discussed in the next section, which mainly comes from realistic constraints, performance, and practicality.

2.1 A Rollout Mechanism

Exact value iteration, policy iteration methods, or their variants tend to lead to overwhelming computation requirements as the state space is extremely large. Compromise formulas are various approximates of DPs (ADPs) [Bertsekas(1995),Bertsekas(1976),Tesauro and Galperin(1996),Boyd et al.(2013),Ulmer(2017), Dreyfus(1960)]. We introduce an approximate method known as the open-loop feedback control (OLFC), which is a rollout mechanism [Bertsekas(1995),Dreyfus(1960)], to solve (3). The core concept of OLFC is to ignore in part the availability of online information and target a more tractable computation. And we have the following lemma [Bertsekas(1995)],

Lemma 1 *Rollout policy always results in improved performance over the corresponding base policy.*

Proof 1 *Please refer to Prop. 6.1.1 and Section 6.5.1 in [Bertsekas(1995)].*

Back to our bid optimization problem, we use the impression feature accessed from online as feedback information to adjust the current and future bid prices in real time and ignore the real budget remnant information. In particular, for the m-th auction, we solve the remaining $(T-m+1)$ number of bid price decision rules $\bar{\pi}_t(\mathbf{x}_t)$ through solving the following problem:

$$\begin{aligned}
\arg\max_{\bar{\pi}_t, t=m,\ldots,T} \quad & \sum_{t=m}^{T} \mathbb{E}\left[v(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))\right] \\
& - \sum_{j=1}^{k} \lambda_j \Bigg\{ \mathbb{E}[\sum_{t=m}^{T} c(\mathbf{x}_t, *)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))] \\
& - C_j \sum_{t=m}^{T} \mathbb{E}\left[v_{a,j}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))\right] \Bigg\} \\
\text{s.t.} \quad & \mathbb{E}[\sum_{t=m}^{T} c(\mathbf{x}_t, *)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))] \leq R_m.
\end{aligned} \tag{4}$$

Based on OLFC, remnant budget $R_m = B - B_m$, where B_m is the cost that has been paid for the past $(m-1)$ auctions, for auction m to auction T is averaged with respect to x_t and randomness in auction outcomes and then constrained. Thus the altered strategy only relates to the impression feature $\mathbf{x}$. At the same time, (4) with different m and B_m have the same form. At each iteration m, we solve (4) for a specific R_m, and then evaluate the bid price through $b_m = \bar{\pi}(\mathbf{x}_m)$, We further simplify the formulation of (4) via the following operations. First, we consider taking expectation with respect to the randomness of auctions and with respect to $\mathbf{x}_t$'s separately. Consequently, given $\mathbf{x}_t$, we have

$$\mathbb{E}[\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))] = \mathbb{P}(\text{Advertiser} \quad \text{wins}|\mathbf{x}_t, \bar{\pi}_t) := q_t(\mathbf{x}_t, \bar{\pi}_t).$$

When $\bar{\pi}_t$ is clearly known from context, we drop it, writing $q_t(\mathbf{x}_t) := q_t(\mathbf{x}_t, \bar{\pi})$ for short. Letting $q_t(\mathbf{x}_t)$ be the function to be determined, we come up with the following optimization problem.

$$\begin{aligned}
\arg\max_{q_t, t=m,\ldots,T} \quad & \sum_{t=m}^{T} \mathbb{E}\left[v(\mathbf{x}_t)q_t(\mathbf{x}_t)\right] \\
\text{s.t.} \quad & \mathbb{E}[c(\mathbf{x}_t, *)q_t(\mathbf{x}_t)] \leq R_m \\
& \frac{\sum_{t=m}^{T} \mathbb{E}[c(\mathbf{x}_t, *)q_t(\mathbf{x}_t)]}{\sum_{t=m}^{T} \mathbb{E}[v(\mathbf{x}_t)q_t(\mathbf{x}_t)]} \leq C_j, \;\; j = 1, \ldots, k.
\end{aligned} \tag{5}$$

Besides, the impression feature $\mathbf{x}$ is i.i.d and draws from the same set $\mathcal{X}$ subject to a PDF $p_{\mathbf{x}}$. It indicates that the quantity of the remnant budget B_m affects the strategy of each auction. If we could solve (5), we would reversely obtain $\bar{\pi}(\mathbf{x}_t)$ from $q_t(\mathbf{x}_t)$. However it is extremely difficult to solve $q_t(\mathbf{x}_t)$ from (5) as $\mathcal{X}$ in general is a continuous set, solving $q_t(\mathbf{x}_t)$ amounts to finding an optimal mapping from $\mathcal{X}$ to $[0,1]$ for (5). To numerically solve (5) in a tractable way, we resort to a sampling method.

3 Implementation trade-off in solving the bid optimization

Though we know the structure of the bidding function (5), it is still difficult to solve. The reasons are not only the high frequency and time-delay mentioned in the implementation challenges, but also the difficulties of lacking the exact knowledge about $w(\mathbf{x})$, $p_{\mathbf{x}}$, and other functions in (5). In this section, we solve the bid pricing rule numerically using historical auction logs, which transforms the problem of finding an optimal mapping q_t of $\mathbf{x}_t$ for (5) into a linear programming problem with decision variable being a set of function values for q_t evaluated at archived impression features in an auction log. Further, we derive an approximate bid price decision rule to the one obtained from solving (4). This is done by analysing the structure of solutions to the linear programming and its dual . In particular, the pricing rule can be easily parameterised using an optimal solution to the dual linear programming problem. It is renewed over time by solving a sequence of receding dual linear programming problems. Finally, practical issues regarding balancing computational complexity and accuracy are also be discussed.

3.1 Solution Analytics from Linear Programming Dual Theory

To come up with decision rules $\bar{\pi}$, we need to have the functions of $p_{\mathbf{x}}$, $v(\mathbf{x})$, and other variables in (5) first. However, they are inaccessible at the start of the day because impressions would not have appeared yet. We have to assume that the impression features follow the same PDF $p_{\mathbf{x}}$ and set $\mathcal{X}$ in different days, which is widely used in previous studies [Han et al.(2017), Zhang(2016), He et al.(2021), Yang et al.(2019)]. Thus, we can approximately identify the variables from the logs. Nevertheless, it is challenging to identify $p_{\mathbf{x}}$ due to the construction of $\mathbf{x}$, a high-dimensional vector containing lots of information (refer to Section 1.1 and [Yuan et al.(2013), Zhang and Xu(2014)]). Some machine learning methods can be used to learn $p_{\mathbf{x}}$ and others [Cai et al.(2017), Jin et al.(2018), Ren et al.(2017)], it may not be suitable for such a huge scale of ad campaigns due to the limited resources in practice as well as the online implementation.

An auction log record is a description of an auctioned impression in the past. Inspired by [Han et al.(2017), Yang et al.(2019)], every day's auction features recorded in logs can be seen as a collection of samples from $\mathcal{X}$. Thus we replace the expectations in (5) with their estimates, generated from logs and solve :

$$\begin{aligned} \mathop{\arg\max}_{\hat{q}_t(\hat{\mathbf{x}}_t), t=m,\ldots,\hat{T}} \quad & \sum_{t=m}^{\hat{T}} v(\hat{\mathbf{x}}_t)\hat{q}_t(\hat{\mathbf{x}}_t) \\ \text{s.t.} \quad & \sum_{t=m}^{\hat{T}} \hat{c}_t \hat{q}_t(\hat{\mathbf{x}}_t) \le R_m \\ & \frac{\sum_{t=m}^{\hat{T}} \hat{c}_t \hat{q}_t(\hat{\mathbf{x}}_t)}{\sum_{t=m}^{\hat{T}} v_{a,j}\hat{q}_t(\hat{\mathbf{x}}_t)} \le C_j, \ j = 1, \ldots, k, \end{aligned} \tag{6}$$

where the superscript $\wedge$ denotes the value recorded in logs. The auction log for an ad campaign is one-day based. All the auction records in a day form an auction log. Note that the total impression T for the current trading day may be different from logged value $\hat{T}$. Besides, as we mentioned in Section 1.1, the billing method used in this paper is click-based. It means that $\hat{c}_t = \hat{w}_t \hat{p}^{\text{CTR}}(\hat{\mathbf{x}}_t)$ when we evaluate $\mathbb{E}[c(\mathbf{x}_t, *)|\mathbf{x}_t]$ with logs, where $\hat{p}^{\text{CTR}}(\hat{\mathbf{x}}_t)$ is the predicted click-through rate of impression $\hat{\mathbf{x}}_t$ recorded in logs.

We analyse the structure of the solution to (6) and have the following lemma.

Lemma 2 *If* (6) *is feasible, then there exists an optimal solution* $\hat{q}_t^*(\mathbf{x}_t), t = m, \ldots, \hat{T}$*, of* (6)*, which has at least* $(\hat{T} - m - k)$ *number of 0 and 1.*

Proof 2 *We rewrite* (6) *to an equivalent constrained linear programming by introducing slack variables to transform the inequality constraints into linear ones:*

$$\begin{aligned} \mathop{\arg\max}_{\hat{q}_t(\hat{\mathbf{x}}_t), a_t, d_j, g} \quad & \sum_{t=m}^{\hat{T}} v(\mathbf{x}_t)\hat{q}_t(\hat{\mathbf{x}}_t) \\ s.t. \ g + & \sum_{t=m}^{\hat{T}} \hat{c}_t \hat{q}_t(\hat{\mathbf{x}}_t) = R_m \\ & \hat{q}_t(\hat{\mathbf{x}}_t) + a_t = 1, \ t = m, \ldots, \hat{T} \\ & d_j + \sum_{t=m}^{\hat{T}} \hat{q}_t(\hat{\mathbf{x}}_t)\left(\hat{c}_t - C_j v_{a,j}(\hat{\mathbf{x}}_t)\right) = 0, \ j = 1, \ldots, k \\ & \hat{q}_t(\hat{\mathbf{x}}_t) \ge 0, a_t \ge 0, \ t = m, \ldots, \hat{T} \\ & d_j \ge 0, \ j = 1, \ldots, k \\ & g \ge 0, \end{aligned} \quad , \tag{7}$$

which contains $(2\hat{T} - 2m + k + 3)$ *variables and* $(\hat{T} - m + k + 2)$ *equality constraints. We can define the so-called basic feasible solutions (BFS) for the form of LP problems like* (7)*. The BFSs of* (7) *have at least* $(\hat{T} - m + 1)$ *number of* 0*. From [Schrijver(1998), Theorem 2.7], if* (6) *is feasible, then* (7) *is feasible and* *there exists an optimal solution that is a BFS for* (7) . *Furthermore, note that, due to the constraint* $\hat{q}_t(\hat{\mathbf{x}}_t) + a_t = 1$, $\forall t = m, \ldots, \hat{T}$, *if* $a_t = 0$ *then* $\hat{q}_t(\hat{\mathbf{x}}_t) = 1$. *Thus, when we only consider* $\hat{q}_m, \ldots, \hat{q}_{\hat{T}}$ *of an optimal BFS as an optimal solution to* (6), *it has at least* $(\hat{T} - m - k)$ *number of* 0 *and* 1. □

We continue to investigate (6) from the dual theory. Slack $\hat{q}$ to $[0, 1]$ and consider the dual problem of (6),

$$
\begin{aligned}
\underset{\alpha,\beta_j,\gamma_t}{\arg\min}\ & R_m\alpha + \sum_{t=m}^{\hat{T}} \gamma_t \\
\text{s.t.}\ \ & \hat{c}_t\alpha + \sum_{j=1}^{k} \left(\hat{c}_t - C_j v_{a,j}(\hat{\mathbf{x}}_t)\right) \beta_j - v(\hat{\mathbf{x}}_t) + \gamma_t \geq 0, \\
& t = m, \ldots, \hat{T}, \\
& \gamma_t \geq 0, t = m, \ldots, \hat{T}, \\
& \beta_j \geq 0,\ j = 1, \ldots, k, \\
& \alpha \geq 0.
\end{aligned}
\tag{8}
$$

Let α^*, $\beta_1^*, \ldots, \beta_k^*$ be the optimal to (8). We denote

$$
\mu_t = \frac{1}{\hat{p}^{\mathrm{CTR}}(\hat{\mathbf{x}}_t)} \frac{v(\hat{\mathbf{x}}_t) + \sum_{j=1}^{k} \beta_j^* C_j v_{a,j}(\hat{\mathbf{x}}_t)}{\alpha^* + \sum_{j=1}^{k} \beta_j^*} \tag{9}
$$

Lemma 3 *Suppose that* (6) *is feasible. Consider the optimal solution* $\hat{q}_t^*(\mathbf{x}_t), t = m \ldots, \hat{T}$, *studied in Lemma 2. Then* $\hat{q}_t^*(\hat{\mathbf{x}}_t) = 0$ *if* $\mu_t < \hat{w}_t$, *and* $\hat{q}_t^*(\hat{\mathbf{x}}_t) > 0$ *if* $\mu_t > \hat{w}_t$.

Proof 3 *Based on the complementary slackness [Bertsimas and Tsitsiklis(1997), Yang et al.(2019)], the optimal solutions of* (8) *and* (7) *satisfy* $\forall t = m, \ldots, \hat{T}$:

$$
\hat{q}_t^*(\hat{\mathbf{x}}_t) \left\{ \hat{c}_t \left(\alpha^* + \sum_{j=1}^{k} \beta_j^* \right) - v(\hat{\mathbf{x}}_t) - \sum_{j=1}^{k} C_j v_{a,j}(\hat{\mathbf{x}}_t)\beta_j^* + \gamma_t^* \right\} = 0, \tag{10}
$$

$$
(\hat{q}_t^* - 1)\gamma_t^* = 0, \tag{11}
$$

where (11) *comes from the complementary slackness of* $\hat{q} \in [0, 1]$. *Then substituting* (9) *and* $\hat{c}_t = \hat{w}_t \hat{p}_t^{\mathrm{CTR}}$ *into* (10),

$$
\hat{q}_t^*(\hat{\mathbf{x}}_t) \left\{ \hat{p}^{\mathrm{CTR}}(\hat{\mathbf{x}}_t) \left(\hat{w}_t - \mu_t\right) \left(\alpha^* + \sum_{j=1}^{k} \beta_j^* \right) + \gamma_t^* \right\} = 0.
$$

If $\mu_t < \hat{w}_t$, *then* $\hat{p}^{\mathrm{CTR}}(\hat{\mathbf{x}}_t) \left(\hat{w}_t - \mu_t\right) \left(\alpha^* + \sum_{j=1}^{k} \beta_j^* \right) + \gamma_t^* > 0$, *which further implies that* $\hat{q}_t^*(\hat{\mathbf{x}}_t) = 0$ *by* (10). *Otherwise, if* $\hat{q}_t^*(\hat{\mathbf{x}}_t) = 0$, *then* $\gamma_t^* = 0$ *from* (11). *Hence* $\mu_t \leq \hat{w}_t$ *according to feasibility of* (8), *which completes the proof.* □

Finally, we have the following theorem.

Theorem 1 *Suppose that $\hat{T} \gg k$ and that* (6) *is feasible. The following bidding rule of* (6)

$$\bar{\pi}_t(\hat{\mathbf{x}}_t) = \frac{1}{\hat{p}^{\mathrm{CTR}}(\hat{\mathbf{x}}_t)} \frac{v(\hat{\mathbf{x}}_t) + \sum_{j=1}^k \beta_j^* C_j v_{a,j}(\hat{\mathbf{x}}_t)}{\alpha^* + \sum_{j=1}^k \beta_j^*}. \tag{12}$$

is optimal in hindsight (It is in hindsight since at impression t the realizations $\hat{\mathbf{x}}_t$ and $\hat{w}_t$ from $t = m$ to $t = \hat{T}$ have already been known for biding decision making.) with respect to the second-price auction mechanism.

Proof 4 *Consider the bidding strategy* (12)*. If $\bar{\pi}_t(\hat{\mathbf{x}}_t) > \hat{w}_t$, then the bidder will win the impression according to the second-price auction mechanism. On the other hand, $\hat{q}_t^*(\hat{\mathbf{x}}_t) > 0$ by Lemma 3. If $\bar{\pi}_t(\hat{\mathbf{x}}_t) < \hat{w}_t$, then the bidder will lose the impression by the second-price auction mechanism. In this case $\hat{q}_t^*(\hat{\mathbf{x}}_t) = 0$ by Lemma 3. In virtue of Lemma 2, when $\hat{q}_t^*(\hat{\mathbf{x}}_t) > 0$ there are at most $(k+1)$ number of $\hat{q}_t^*(\hat{\mathbf{x}}_t)$ that is not 1. Noticing that $\hat{T} \gg k$, it concludes the proof.* □

The assumption $\hat{T} \gg k$ is practical in most cases as the number of real ad campaigns is mostly around or greater than the order of 10^4. In contrast, the number of CPA constraints is at most one or two in most cases [Yuan et al.(2014),Zhang(2016),Yang et al.(2019),Google.com(2022),Han et al.(2017)]. Thus, for a batch of $\hat{T}$ impressions, when we compare auction results that come from an LP solver 2 and Theorem 1, the number of different results would not exceed $k+1$, which is an extremely small quantity compared to the total number of auctions $\hat{T}$.

We use the rule (12) as an approximate to the bid pricing rule obtained from solving (4) for the ongoing impression auction. Thus a bid price for the current auction can be proposed by valuing the current bid price using the current $\mathbf{x}_t$ via (12), that is,

$$\bar{\pi}_t(\mathbf{x}_t) = \frac{1}{p_t^{\mathrm{CTR}}(\mathbf{x}_t)} \frac{v(\mathbf{x}_t) + \sum_{j=1}^k \beta_j^* C_j v_{a,j}(\mathbf{x}_t)}{\alpha^* + \sum_{j=1}^k \beta_j^*}. \tag{13}$$

3.2 Auctions and Bid Requests Aggregation

In practice, we may not exactly know the remnant budget at the start of each auction since the expense for each auction is in connection with users' clicks, which causes time delay [Yang et al.(2019), Zhang(2016)]. A possible solution is to aggregate impressions within a period of time and then fix $\bar{\pi}$ in each period [Yang et al.(2019), Zhang(2016), Han et al.(2017)]. At the end of the period, the remnant budget is re-evaluated via the billing module. After that, we can renew $\bar{\pi}$ in preparation for the next aggregation of impressions.

4 Supplements for down-sampling

We are do working on the strictly theoretical proof for the down-sampling process, and the rules for choosing a reasonable sampling size. It is challenging because it relates to many areas, such as sampling theory, confidence, dual linear programming, etc. But, we have made a series of experiments to verify the effectiveness of down-sampling and provided some empirical for choosing the sampling size.

We randomly select 1000 ad campaigns from Taobao on the same day. Among them, the minimal number of auctions is less than 10^4; while the maximum is over 10^6. The sampling size is defined as N, and we set five different $N = \{1000, 5000, 10000, 20000, 30000\}$ for experiments. Each campaign was sampled and solved by 100 times for each N. The experiments contain three parts :

(1) For BCB, accuracy loss and stability of solution in the bidding formulation are subject to different sample sizes N. For MCB, comparing parameters is futile because more than one parameter is used. Thus, we test bid price accuracy loss and stability subject to different N.

(2) For BCB, accuracy loss and stability of solution in the bidding formulation are subject to different sample ratios. Also, for MCB, we test bid price accuracy loss and stability subject to different sample ratios.

(3) Consumed computation resources subject to different N.

Compared metrics are defined in the following:

(1) For each campaign in BCB, we use α_N to denote solutions come from different N. And we regard the solution solved by using the whole auction log as the exact, denoted by α^*. Similarly, for each campaign in MCB, we use b_N and b^* to denote bids from different N and the exact solution by the whole auction log, respectively. Then define a normalized metric

$$\tilde{\alpha}_N = \alpha_N/\alpha^* - 1, \quad \tilde{b}_N = b_N/b^* - 1 \tag{14}$$

where $\tilde{\alpha}_N$ and $\tilde{b}_N$ are the closer to 0, the better.

(2) Our system is implemented on Dataworks, an end-to-end big data development and governance platform [Cloud(2022)]. The used solver is MindOpt [Academy(2022)]. To eliminate gaps among different implementation environments, we set resources used to solve $N = 1000$ as the baseline. Then defined

$$\tilde{T}_N = \frac{T_N}{T_{1000}},\ \tilde{C}_N = \frac{C_N}{C_{1000}},\ \tilde{M}_N = \frac{M_N}{M_{1000}}, \tag{15}$$

where for a sampling size N, T_N, C_N, and M_N are the consumed time, CPU resources, and memory resources, respectively.

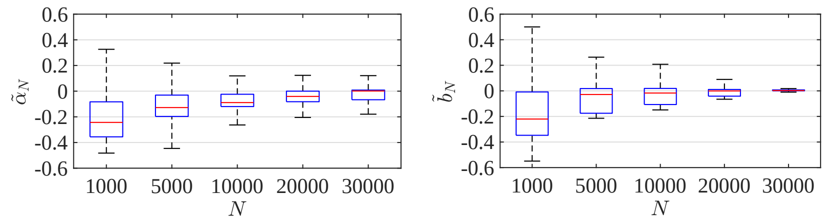

Figure 1: The relative performance loss of solutions with respect to different sampling sizes N. The left is about BCB, $\tilde{\alpha}_N$ v.s. N. The right is about MCB, $\tilde{b}_N$ v.s. N.

The first comparison is plotted in Fig. 1. We use boxplots to represent the solution performance loss to different N. In the second comparison, the sampling ratio is discrete and non-uniform, and the normalized metrics jump around zero, which makes the results unclear.The absolute values of $\tilde{\alpha}_N$ and $\tilde{b}_N$, and then Fig. 2 shows their upper quantiles' trending. Finally, the comparison of computation resources is Fig. 3, which takes the average of whole campaigns. To sum up:

(1) The accuracy and stability of solutions are related to N. Increasing N can improve its accuracy and stability but exponentially increases computational resources simultaneously. This rule is the same for increasing the sampling ratio.

(2) Solution improvements are not infinite as increasing N or the sampling ratio. When N is greater than 10^4, improvements are less significant, but the required computation resources are overgrowing. Also, for the sampling ratio, it can achieve 90% accuracy when it exceeds 3% and reaches 95% with 5% of the data in logs.

(3) Setting the sampling size to 10000 may be suitable for the usage. Also, for each ad campaign, we also can use $3\% \sim 5\%$ of the data in its auction log for a personalized option.

5 Stability of impression distribution

The assumption of a stable distribution of feature $\mathbf{x}$ and corresponding values is vital for solving bid optimization [Yuan et al.(2013), Zhang et al.(2014), Yang et al.(2019), Guan et al.(2021)]. However, it is hard to study $\mathbf{x}$'s distribution, a high-dimensional vector defined by different platforms [Yuan et al.(2013), Zhang and Xu(2014)]. Thus, we only can verify the mapped values, that is, $w, p^{\mathrm{CTR}}, p^{\mathrm{CVR}}, p^{\mathrm{WCVR}}$.

Similarly, we randomly select 1000 ad campaigns throughout Sep. 2022, whose data is nearly 1 TB. Then comparing each value's CDFs, and conducting

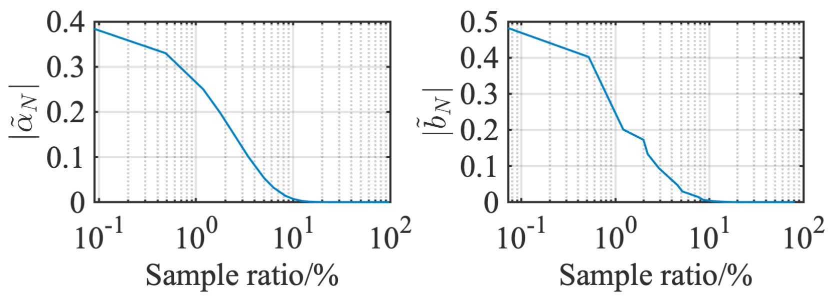

Figure 2: The absolute performance loss of solutions with respect ot different sample ratios. The left is about BCB, the right is about MCB.

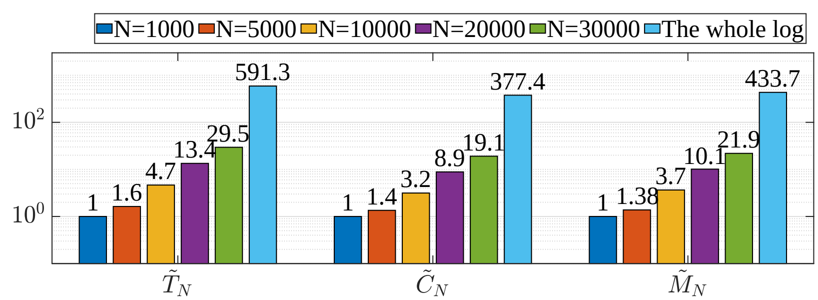

Figure 3: Computation resources v.s. sampling size N. The left is the normalized computation time. The middle is the normalized consumed CPU. The right is the normalized consumed memory.

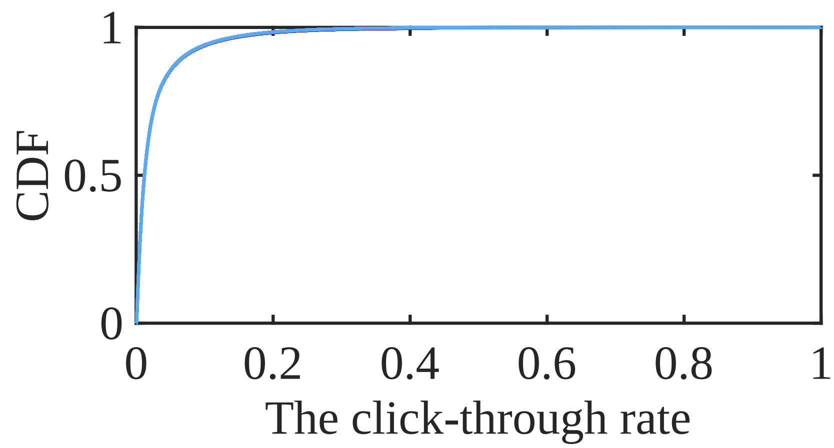

Figure 4: The cumulative distribution functions of the click-through rate in different days throughout Sep. 2022.

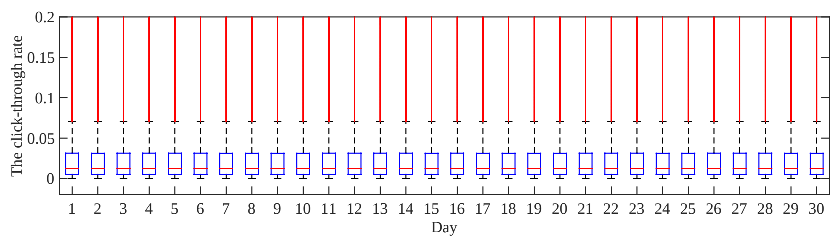

Figure 5: Boxplots of the click-through rate in different days throughout Sep. 2022. The red dots outside the blue box are outliers. It seems to be the red lines due to too many outliers.

K-S tests [Massey Jr(1951), Fasano and Franceschini(1987)]. For example, we plot CDFs of each day's CTR in Fig.4, where 30 trajectories are nearly identical and overlap. Meanwhile, boxplots of CTR in the different days are shown in Fig. 5. The results for the rest three values are the same and will not be repeated here. Further, we regard Sep. 1st, 2022, as the base. The K-S test shows that the distribution of these four values in the following 29 days is statistically consistent with the ground with 95% confidence at least. As a result, they can be considered to come from the same distribution.

6 Auction scale prediction details

Firstly, we divide a day into n_τ intervals, following the update interval set by our method. Then, the module predicts the expected auction scales of each interval in the future. To do so, one is for the training convergence because the

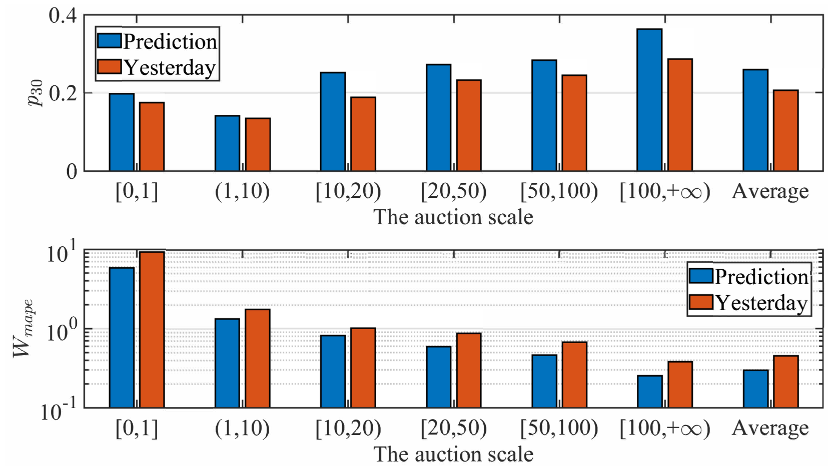

Figure 6: The comparison of p_{30} and w_{mape}. The last bar is the average of the whole test set.

auction scale varies largely with different campaigns in a day, from 0 to over a million, which may lead to a scant reward and state construction. On the other hand, the auction scale in a day may be time-related among different intervals. Separating a day into several intervals may reveal this relationship.

A fully connected network is used in the prediction module, which has two layers with 256 neurons in each layer. The activation function is *tanh*. Soft attention with one neuron is used in attention pooling. A 128-dimensional vector, which consists of and encodes the predicted ad campaign features, the auction scales of each past interval, and the time feature, inputs the network. The output is the predicted auction scales for each interval in the future.

We conducted an offline experiment to verify the effectiveness of the prediction module compared to directly using yesterday's auction scale. We use the auction logs from 11th May 2022 to 17th May 2022 provided by Taobao for network training, whose set is over 500 GB of data and over 5 million ad campaigns. Then we use the module to predict the auction scale of each campaign in 18th May 2022 and 19th May 2022, which also has over 5 million campaigns. Though a day is divided into several intervals, the auction scales still vary significantly for different test data. Thus we set several groups based on the auction scales to compare the prediction we provided and the one that directly uses yesterday's data. There are six groups, which are $[0, 1]$, $(1, 10)$, $(10, 20)$, $(20, 50)$, $(50, 100)$, and $(100, +\infty)$. For example, $[0, 1]$ means that the auction scale is less than/equals to 1 time; $(20, 50)$ means the auction scale is greater than 20 times but less than 50. Finally, the compared metrics are:

- **The probability of the absolute error less than** 30%, denoted by p_{30}, the higher, the better [Nath et al.(2013)]. It means the probability of the prediction is around the truth $\pm 30\%$.

- **The weighted mean absolute percentage error**, denoted by W_{mape}, the lower, the better. In this problem, each prediction error has the same weight.

The results are shown in Fig. 6. As desired, the prediction is difficult for those with few impressions campaigns, and the corresponding error is big. Meanwhile, when the auction scale increases, the accuracy p_{30} increases, and the absolute error w_{mape} decreases. Compared to the approach that directly uses yesterday's data, the designed prediction module is effective and achieves better performance by at least 20%. Though its prediction still has a significant gap to the truth, it is enough for use in bid optimization [Yuan et al.(2013), Nath et al.(2013), Zhang et al.(2014)]. Also, it is an improvement direction of bid optimization in the future.

References

[Academy(2022)] Alibaba Damo Academy. 2022. MindOpt Optimization Suite.

[Bertsekas(1976)] Dimitri P Bertsekas. 1976. Dynamic programming and stochastic control. (1976).

[Bertsekas(1995)] D. P. Bertsekas. 1995. Dynamic Programming and Optimal Control. *Athena Scientific,* (1995).

[Bertsimas and Tsitsiklis(1997)] Dimitris Bertsimas and John N Tsitsiklis. 1997. *Introduction to linear optimization.* Vol. 6. Athena Scientific Belmont, MA.

[Boyd et al.(2013)] Stephen Boyd, Mark T Mueller, Brendan O'Donoghue, Yang Wang, et al. 2013. Performance bounds and suboptimal policies for multi–period investment. *Foundations and Trends® in Optimization* 1, 1 (2013), 1–72.

[Cai et al.(2017)] 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.* 661–670.

[Cloud(2022)] Alibaba Cloud. 2022. DataWorks.

[Dreyfus(1960)] Stuart E Dreyfus. 1960. Dynamic programming and the calculus of variations. *Journal of mathematical analysis and applications* 1, 2 (1960), 228–239.

[Fasano and Franceschini(1987)] Giovanni Fasano and Alberto Franceschini. 1987. A multidimensional version of the Kolmogorov–Smirnov test. *Monthly Notices of the Royal Astronomical Society* 225, 1 (1987), 155–170.

[Google.com(2022)] Google.com. 2022. Google Ads Help.

[Grislain et al.(2019)] Nicolas Grislain, Nicolas Perrin, and Antoine Thabault. 2019. Recurrent neural networks for stochastic control in real-time bidding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2801–2809.

[Guan et al.(2021)] Ziyu Guan, Hongchang Wu, Qingyu Cao, Hao Liu, Wei Zhao, Sheng Li, Cai Xu, Guang Qiu, Jian Xu, and Bo Zheng. 2021. Multi-Agent Cooperative Bidding Games for Multi-Objective Optimization in e-Commercial Sponsored Search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2899–2909.

[Guo et al.(2017)] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[Han et al.(2017)] Z. Han, J. Jin, T. Chang, P. Fei, and K. Gai. 2017. Optimized Cost per Click in Taobao Display Advertising. In *the 23rd ACM SIGKDD International Conference*.

[He et al.(2021)] Yue He, Xiujun Chen, Di Wu, Junwei Pan, Qing Tan, Chuan Yu, Jian Xu, and Xiaoqiang Zhu. 2021. A Unified Solution to Constrained Bidding in Online Display Advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2993–3001.

[Jin et al.(2018)] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. 2018. Real-Time Bidding with Multi-Agent Reinforcement Learning in Display Advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy). Association for Computing Machinery, New York, NY, USA, 2193–2201.

[Karlsson(2020)] N. Karlsson. 2020. Feedback Control in Programmatic Advertising: The Frontier of Optimization in Real-Time Bidding. *IEEE control systems* 40, 5 (2020), 40–77.

[Li and Guan(2014)] Xiang Li and Devin Guan. 2014. Programmatic buying bidding strategies with win rate and winning price estimation in real time mobile advertising. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 447–460.

[Lyu et al.(2022)] Fuyuan Lyu, Xing Tang, Huifeng Guo, Ruiming Tang, Xiuqiang He, Rui Zhang, and Xue Liu. 2022. Memorize, factorize, or be naïve: Learning optimal feature interaction methods for ctr prediction. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1450–1462.

[Massey Jr(1951)] Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association* 46, 253 (1951), 68–78.

[Nath et al.(2013)] Abhirup Nath, Shibnath Mukherjee, Prateek Jain, Navin Goyal, and Srivatsan Laxman. 2013. Ad impression forecasting for sponsored search. In *Proceedings of the 22nd international conference on World Wide Web*. 943–952.

[Ren et al.(2017)] Kan Ren, Weinan Zhang, Ke Chang, Yifei Rong, Yong Yu, and Jun Wang. 2017. Bidding machine: Learning to bid for directly optimizing profits in display advertising. *IEEE Transactions on Knowledge and Data Engineering* 30, 4 (2017), 645–659.

[Schrijver(1998)] Alexander Schrijver. 1998. *Theory of linear and integer programming*. John Wiley & Sons.

[Shih et al.(2020)] Wen-Yueh Shih, Yi-Shu Lu, Hsiao-Ping Tsai, and Jiun-Long Huang. 2020. An expected win rate-based real time bidding strategy for branding campaign by the model-free reinforcement learning model. *IEEE Access* 8 (2020), 151952–151967.

[Tesauro and Galperin(1996)] Gerald Tesauro and Gregory Galperin. 1996. Online policy improvement using Monte-Carlo search. *Advances in Neural Information Processing Systems* 9 (1996).

[Tunuguntla and Hoban(2021)] Srinivas Tunuguntla and Paul R Hoban. 2021. A near-optimal bidding strategy for real-time display advertising auctions. *Journal of Marketing Research* 58, 1 (2021), 1–21.

[Ulmer(2017)] Marlin Wolf Ulmer. 2017. *Approximate dynamic programming for dynamic vehicle routing*. Vol. 1. Springer.

[Vickrey(1961)] William Vickrey. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance* 16, 1 (1961), 8–37.

[Wang et al.(2016)] Yuchen Wang, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu. 2016. Functional bid landscape forecasting for display advertising. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 115–131.

[Wen et al.(2019)] Hong Wen, Jing Zhang, Quan Lin, Keping Yang, and Pipei Huang. 2019. Multi-level deep cascade trees for conversion rate prediction in recommendation system. In *proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 338–345.

[Wen et al.(2021)] Hong Wen, Jing Zhang, Fuyu Lv, Wentian Bao, Tianyi Wang, and Zulong Chen. 2021. Hierarchically modeling micro and macro behaviors via multi-task learning for conversion rate prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2187–2191.

[Wu et al.(2018)] Wush Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2018. Deep censored learning of the winning price in the real time bidding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2526–2535.

[Yang et al.(2019)] Xun Yang, Yasong Li, Hao Wang, Di Wu, Qing Tan, Jian Xu, and Kun Gai. 2019. Bid optimization by multivariable control in display advertising. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1966–1974.

[Yuan et al.(2013)] 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*. 1–8.

[Yuan et al.(2014)] Yong Yuan, Feiyue Wang, Juanjuan Li, and Rui Qin. 2014. A survey on real time bidding advertising. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*. IEEE, 418–423.

[Zhang(2016)] Weinan Zhang. 2016. *Optimal real-time bidding for display advertising*. Ph. D. Dissertation. UCL (University College London).

[Zhang et al.(2021)] Weinan Zhang, Jiarui Qin, Wei Guo, Ruiming Tang, and Xiuqiang He. 2021. Deep learning for click-through rate estimation. *arXiv preprint arXiv:2104.10584* (2021).

[Zhang and Xu(2014)] Weinan Zhang and Jian Xu. 2014. Learning, Prediction and Optimisation in RTB Display Advertising. In *Proceedings of the 25th Information and Knowledge Management*. 1077–1086.

[Zhang et al.(2014)] 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*. 1077–1086.

[Zhou et al.(2018)] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[Zhu et al.(2017)] Wen-Yuan Zhu, Wen-Yueh Shih, Ying-Hsuan Lee, Wen-Chih Peng, and Jiun-Long Huang. 2017. A gamma-based regression for winning price estimation in real-time bidding advertising. In *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 1610–1619.