Supplement

I. 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.

A. Technical terms and notations

First, some technical terms and notations are explained; refer to [1], [2] 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 $p^{\mathrm{CTR}} \in [0, 1]$. It is the estimated probability that users click through an ad when they see it.
- **Conversion rate, CVR**, denoted by $p^{\mathrm{CVR}} \in [0, 1]$. 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 [2], [3]. 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 [4], 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 [2]. 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 [3], [5].

(c) Bid optimization problem-related terms and notations.
- **Key performance indicator, KPI**. It is a quantitative measurement of advertising performance [3] 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 $w = w(\mathbf{x})$ [2]. 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 [2], [3], [6]. 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 $\mathbf{x}$ [7]. Besides, the billing method is another factor affecting the cost [1]. 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 [5]; while in display ads, billing has nothing to do with whether an ad has been clicked or not [8]. 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.

B. 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 [6], 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 [4], [9], [10], 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

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

where $j = 1, \dots, k$, indicator function $\mathbf{1}(w(\mathbf{x}_t), \pi_t(B_t, \mathbf{x}_t))$ means the t-th auction result, if $\pi_t(B_t, \mathbf{x}_t) \ge w(\mathbf{x}_t)$, 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 $\mathbf{x}_t$.The objective in the above is to find $\pi_t, t \in \{1, \dots, T\}$ 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(see Section **??**), such as cost per click, cost per conversion. etc.

1) A real-world example:

As an example, we present a real-world business service [5] provided by *Taobao.com* to illustrate the functional optimization problem (1). Its main settings contain:

- **Maximizing objective**, $v(\mathbf{x}) = p^{\mathrm{CTR}}(\mathbf{x})p^{\mathrm{CVR}}(\mathbf{x})$, 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 $v_a(\mathbf{x}) = p^{\mathrm{CTR}}(\mathbf{x})$. 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

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

C. 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:

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

  The sequence of $\mathbf{x}_t$ is an i.i.d. random process (see Section I-A), that is, the transition of $\mathbf{x}_{t+1}$ is not affected by action b_t and 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 a bit complicated because of the punishment of constraints,

  $$\begin{aligned} r_t = v(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \pi(s_t)) - \sum_{j=1}^{k}\lambda_j \\ \{\mathbb{E}\left[b_t\right] - C_j\mathbb{E}\left[v_{a,j}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \pi(s_t))\right]\} \end{aligned}$$

  where $\lambda_j > 0, j \in \{1, \dots, k\}$ 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 a bid optimization problem. Though they are important in RTB, their algorithms are beyond the scope of this paper. In a real DSP, as shown in Fig.**??**, these functions mainly come from feature mapping and value estimation modules, such as CTR prediction [11], [12], CVR prediction [13], [14], winning price and winning

rate prediction [15]–[18], landscape prediction [19], feature construction [6], and so on. Thus, the bid optimization in MDP model is to solve:

$$\begin{aligned} &\arg\max_{\pi_t, t=1,\dots,T} \sum_{t=1}^{T} \mathbb{E}\left[r_t\right] \\ &\text{s.t.} \sum_{t=1}^{T} b_t \leq B, \end{aligned} \tag{3}$$

It is worth pointing out that, in most previous optimization-based works [2], [5], [8], [20], [21], researchers do not consider the influence of 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, some machine-learning-based works [22]–[26], budget dynamics has been considered in MDP modelling. Nevertheless, they need enormous resources and data to train 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.

II. 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 a bidding strategy mapping the feature, among others, to a bid price. .
(c) In Table. **??**, 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 demands on infrastructure and algorithms.
(d) The rewards and transition of the consumed budget are time-delayed [3]. As the example in Section I-B1, 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 will be discussed in the next section, which mainly comes from realistic constraints, performance and practicality.

A. 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 approximate DPs (ADPs) [27]–[32]. We will introduce an approximate method known as the open-loop feedback control(OLFC), which is a rollout mechanism [27], [32], to solve (3). The core idea of OLFC is to ignore in part the availability of online information but to target a more tractable computation. Back to our bid optimization problem, we will use the impression feature to be access 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 controller $\bar{\pi}_t(\mathbf{x}_t)$ via solving the following problem:

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

where $V_{a,j}(m-1) = \sum_{t=1}^{m-1} v_{a,j}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), b_t)$ is the amount of impression actions has received in the past auctions. Based on OLFC, B_t transforms to a dynamic constraint (4) in the sense of average, and the 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 shall solve (4) for a specific $R_m = B - B_m$, and then let $b_m = \bar{\pi}(\mathbf{x}_m)$, where B_m is the cost that has been paid for the past $(m-1)$ auctions.

We shall further simplify the formulation of (4) via the following operations. First we consider taking executions with respect to the randomness of auctions and with respect to $\mathbf{x}_t$'s separately. As a consequence, given $\mathbf{x}_t$ we have

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

When $\bar{\pi}_t$ is clearly known from context, we will 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.

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

Besides, the impression feature $\mathbf{x}$ is i.i.d and comes from the same set $\mathcal{X}$ subject to a PDF $p_{\mathbf{x}}$. It indicates that the remnant budget is the fundamental thing that 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 difficulty to solve $q_t(\mathbf{x}_t)$ from (5) as $\mathcal{X}$ in general is a continuous set, solving $q_t(\mathbf{x}_t)$ from (5) 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.

III. Implementation trade-off in solving the bid optimization

Though we know the structure of the bidding function, 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). Thus, in this section, we first introduce two simplifications that make (5) solvable in reality and then an approximate solution using the sampling method is proposed to balance the efficiency and accuracy of the DSP.

A. Solution Analytics from Linear Programming Dual Theory

To get the bidding strategy, 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 the impression features follow the same PDF $p_{\mathbf{x}}$ and set $\mathcal{X}$ in different days, which is widely used in previous studies [2], [5], [8], [24]. 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 I-A and [3], [6]). Some machine learning methods can be used to learn $p_{\mathbf{x}}$ and others [22], [25], [26], it may not be suitable for such a huge scale of ad campaigns (refer to Fig. **??**) due to the limited resources in practice as well as the online implementation.

Inspired by [5], [8], every day's auctions can be seen as collections that are uniformly sampled from $\mathcal{X}$, and their information are recorded in logs. Thus, we can replace those random variables by the records, and then solve (5) to find strategies $\bar{\pi}_t, t \in \{m, \dots, \hat{T}\}$, which can maximize:

$$\begin{aligned} \max \quad & \sum_{t=m}^{\hat{T}} \hat{v}(\hat{\mathbf{x}}_t)\mathbf{1}(\hat{w}_t, \bar{\pi}_t(\hat{\mathbf{x}}_t)) \\ \text{s.t.} \quad & B_{\hat{T}+1}^{\hat{m}} \le B \\ & \frac{B_{\hat{T}+1}^{\hat{m}}}{\hat{V}_{a,j}(m-1) + \sum_{t=m}^{\hat{T}} \hat{v}_{a,j}(\hat{\mathbf{x}}_t)\mathbf{1}(\hat{w}_t, \bar{\pi}_t(\hat{\mathbf{x}}_t))} \le C_j, \end{aligned} \tag{6}$$

where the superscript $\wedge$ denotes the recorded value, and

$$\hat{V}_{a,j}(m-1) = \sum_{t=1}^{m-1} \hat{v}_{a,j}(\hat{\mathbf{x}}_t)\mathbf{1}(\hat{w}_t, \bar{\pi}_t(\hat{\mathbf{x}}_t)), \forall j = 1, \dots, k$$

$$B_{\hat{T}+1}^{\hat{m}} = B_m + \sum_{t=m}^{\hat{T}} \hat{w}_t \rho(*, \hat{\mathbf{x}}_t)\mathbf{1}(\hat{w}_t, \bar{\pi}_t(\hat{\mathbf{x}}_t)),$$

where $\rho(*, \hat{\mathbf{x}}_t)$ is a billing factor relates to the billing method and impression features. Actually, $\hat{w}_t\rho(*, \hat{\mathbf{x}}_t)$ is the cost c_t, while it is complicated according to different billing methods, and we cannot use its records $\hat{c}_t$. For example:

(a) In display ads, the cost is deterministic that paying as soon as advertisers win the auction. Thus, $c = \hat{w}$ due to the second-price auction.

(b) In click ads, the cost is stochastic due to the random click-through process. Thus we have nothing alternative to use the expectation $c = \hat{w}p^{\mathrm{CTR}}$ such that solving the problem in the expected sense.

Thus, we define the billing factor ρ to hold the generality. Moreover, the following provides a bidding formulation based on the linear programming dual theory.

Theorem 1: For the approximation problem (6) based on logs, one of the feasible bidding strategies is

$$\bar{\pi}_t(\hat{\mathbf{x}}_t) = \frac{1}{\rho(*, \hat{\mathbf{x}}_t)} \frac{\hat{v}(\hat{\mathbf{x}}_t) + \sum_{j=1}^{k} \beta_j^* C_j \hat{v}_{a,j}(\hat{\mathbf{x}}_t)}{\alpha^* + \sum_{j=1}^{k} \beta_j^*} \tag{7}$$

where $\alpha^*,\ \beta_1^*, \dots, \beta_k^*$ are the optimal dual solutions of an linear programming stemmed from (6).

Proof 1: Noted that, problem (6) is nonlinear programming due to strategies $\bar{\pi}_t$ not directly affecting the objective but through the indicator function. Nevertheless, if we stand from the other perspective, that is, programming which impressions we should win to maximize the objective under constraint. It is an linear programming concerning indicator functions and then constructing a bidding strategy to meet these allocation results. However, are they equivalent? Following the idea, we construct such a strategy and prove the gap between these two idea is small enough to omit.

First, we set $m = 1$ in (6) for convenience and derive the strategy based on [8]. Slack $\mathbf{1}_t \in [0, 1]$, and write dual problem

$$\begin{aligned} \min_{\alpha, \beta_j, \gamma_t} \quad & B\alpha + \sum_{t=1}^{\hat{T}} \gamma_t \\ \text{s.t.} \quad & \forall t = 1, \dots, \hat{T};\ j = 1, \dots, k \\ & \hat{w}_t \rho(*, \hat{\mathbf{x}}_t)\alpha + \sum_{j=1}^{k} \left[\hat{w}_t \rho(*, \hat{\mathbf{x}}_t) - C_j \hat{v}_{a,j}(\hat{\mathbf{x}}_t)\right] \beta_j \ge \hat{v}(\hat{\mathbf{x}}_t) \\ & (\mathbf{1}_t - 1)\gamma_t \le 0 \\ & \alpha, \beta_j, \gamma_t \ge 0, \end{aligned}$$

Thus, based on the complementary slackness, the optimal $\mathbf{1}_t^*$ must satisfy:

$$\begin{aligned} & \forall t = 1, \dots, \hat{T};\ j = 1, \dots, k \\ & \mathbf{1}_t^* \left\{ \hat{v}(\hat{\mathbf{x}}_t) - \hat{w}_t \rho(*, \hat{\mathbf{x}}_t)\alpha - \sum_{j=1}^{k} \left[\hat{w}_t \rho(*, \hat{\mathbf{x}}_t) - C_j \hat{v}_{a,j}(\hat{\mathbf{x}}_t)\right] \beta_j - \gamma_t \right\} \\ & = 0 \end{aligned} \tag{8}$$

Then construct the strategy as follows:

$$\bar{\pi}_t(\hat{\mathbf{x}}_t) = \frac{1}{\rho(*, \hat{\mathbf{x}}_t)} \frac{\hat{v}(\hat{\mathbf{x}}_t) + \sum_{j=1}^{k} \beta_j^* C_j \hat{v}_{a,j}(\hat{\mathbf{x}}_t)}{\alpha^* + \sum_{j=1}^{k} \beta_j^*}$$

Such that rewrite (8),

$$\mathbf{1}_t^* \left\{ \left[\bar{\pi}_t(\hat{\mathbf{x}}_t) - \hat{w}_t\right] \left[\alpha^* + \sum_{j=1}^{k} \beta_j^*\right] - \gamma_t \right\} = 0$$

where if $\bar{\pi}_t(\hat{\mathbf{x}}_t) \leq \hat{w}_t$, we win, $\mathbf{1}_t^* = 1$ and there exists a $\gamma_t \leq 0$ such that the above equals to 0. Otherwise, $\bar{\pi}_t(\hat{\mathbf{x}}_t) < \hat{w}_t$, we loss, $\mathbf{1}_t^* = 1$ and the above still equals to 0. Thus, the constructed $\bar{\pi}$ is a feasible strategy for the problem and allocates $\mathbf{1}$ into 1 and 0.

However, in solving the dual problem, we have slacked $\mathbf{1}_t \in [0, 1]$. There may exist a gap to the original indicator function $\mathbf{1}_t \in \{0, 1\}$. Thus, we derive the optimal linear programming solution structure of (6) concerning $\mathbf{1}$. Rewrite it to an equality constrained optimization problem as follows:

$$
\begin{aligned}
\max_{\mathbf{1}_t, a_t, f, b_t} \quad & \sum_{t=1}^{\hat{T}} \mathbf{1}_t \hat{v}(\hat{x}_t) \\
\text{s.t.} \quad & \sum_{t=1}^{\hat{T}} \hat{w}_t \rho(*, \hat{x}_t) + f = B \\
& \forall t = 1, \ldots, \hat{T}; j = 1, \ldots, k, \\
& \sum_{t=1}^{\hat{T}} \mathbf{1}_t \left[\hat{w}_t \rho(*, \hat{x}_t) - C_j \hat{v}_{a,j}(\hat{\mathbf{x}}_t) \right] + b_j = 0 \\
& \mathbf{1}_t + a_t = 1, \mathbf{1}_t, f, a_t, b_t \geq 0
\end{aligned}
$$

where there are $2\hat{T}+k+1$ variables and $\hat{T}+k+1$ constraints, and its solution is a basic solution; further, we can define the basic feasible solution (BFS) if all of the elements of a basic solution are greater than or equal to 0. And for any BFS, it has at least $(n-m)$ number of zeros, where n and m are the number of variables and constraints, respectively [33]. Such that, in the above problem, there are at least $\hat{T}$ zeros in its solution. On the other hand, $\mathbf{1}_t + \gamma_t = 1$, meaning that there are at least $(\hat{T}-k-1)$ zeros if we count the number of $\mathbf{1}_t$ and γ_t. Consequently, someone use the simplex method to solve the (6) would get a BFS, which means that there exists at least $(\hat{T} - k - 1)$ number of 0 and 1 in the optimal variables $\mathbf{1}$.

To sum up, if there are $\hat{T}$ auctions with k CPA constraints, the difference between $\bar{\pi}$ and linear programming of $\mathbf{1}$ would not exceed $(k+1)$ auctions. And in reality, $\hat{T} \sim 10^5$ but $k \leq 2$, that is, $\hat{T} \gg k$. Thus the gap is acceptable.

B. Auctions and Bid Requests Aggregation

In reality a DSP usually sets an interval t_w, which is a short time horizon compared to a natural day, specifically Δt minutes [2], [5], [8]. We can choose a smaller Δt when the impression is peaking and choose a larger Δt during an impression trough. The total number of intervals t_w in a day is n_w. Further, the bidding strategy only changes at the end of each interval; the consumed budget and the number of actions $V_{a,j}$, $j \in \{1, \ldots, k\}$ that have been received are known [2], [24].

REFERENCES

[1] Google.com, "Google ads help," 2022. [Online]. Available: https://support.google.com/google-ads

[2] W. Zhang, "Optimal real-time bidding for display advertising," Ph.D. dissertation, UCL (University College London), 2016.

[3] S. Yuan, J. Wang, and X. Zhao, "Real-time bidding for online advertising: measurement and analysis," in *Proceedings of the seventh international workshop on data mining for online advertising*, 2013, pp. 1–8.

[4] W. Zhang, J. Qin, W. Guo, R. Tang, and X. He, "Deep learning for click-through rate estimation," *arXiv preprint arXiv:2104.10584*, 2021.

[5] Z. Han, J. Jin, T. Chang, P. Fei, and K. Gai, "Optimized cost per click in taobao display advertising," in *the 23rd ACM SIGKDD International Conference*, 2017.

[6] W. Zhang and J. Xu, "Learning, prediction and optimisation in rtb display advertising," in *Proceedings of the 25th Information and Knowledge Management*, 2014, pp. 1077–1086.

[7] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of finance*, vol. 16, no. 1, pp. 8–37, 1961.

[8] X. Yang, Y. Li, H. Wang, D. Wu, Q. Tan, J. Xu, and K. Gai, "Bid optimization by multivariable control in display advertising," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1966–1974.

[9] Y. Yuan, F. Wang, J. Li, and R. Qin, "A survey on real time bidding advertising," in *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*. IEEE, 2014, pp. 418–423.

[10] F. Lyu, X. Tang, H. Guo, R. Tang, X. He, R. Zhang, and X. Liu, "Memorize, factorize, or be naïve: Learning optimal feature interaction methods for ctr prediction," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 1450–1462.

[11] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.

[12] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1059–1068.

[13] H. Wen, J. Zhang, F. Lv, W. Bao, T. Wang, and Z. Chen, "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*, 2021, pp. 2187–2191.

[14] H. Wen, J. Zhang, Q. Lin, K. Yang, and P. Huang, "Multi-level deep cascade trees for conversion rate prediction in recommendation system," in *proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 338–345.

[15] W. Wu, M.-Y. Yeh, and M.-S. Chen, "Deep censored learning of the winning price in the real time bidding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2526–2535.

[16] X. Li and D. Guan, "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, 2014, pp. 447–460.

[17] W.-Y. Shih, Y.-S. Lu, H.-P. Tsai, and J.-L. Huang, "An expected win rate-based real time bidding strategy for branding campaign by the model-free reinforcement learning model," *IEEE Access*, vol. 8, pp. 151 952–151 967, 2020.

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

[19] Y. Wang, K. Ren, W. Zhang, J. Wang, and Y. Yu, "Functional bid landscape forecasting for display advertising," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 115–131.

[20] N. Karlsson, "Feedback control in programmatic advertising: The frontier of optimization in real-time bidding," *IEEE control systems*, vol. 40, no. 5, pp. 40–77, 2020.

[21] S. Tunuguntla and P. R. Hoban, "A near-optimal bidding strategy for real-time display advertising auctions," *Journal of Marketing Research*, vol. 58, no. 1, pp. 1–21, 2021.

[22] H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo, "Real-time bidding by reinforcement learning in display advertising," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 661–670.

[23] N. Grislain, N. Perrin, and A. Thabault, "Recurrent neural networks for stochastic control in real-time bidding," in *Proceedings of the 25th*

ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2801–2809.

[24] Y. He, X. Chen, D. Wu, J. Pan, Q. Tan, C. Yu, J. Xu, and X. Zhu, "A unified solution to constrained bidding in online display advertising," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2993–3001.

[25] J. Jin, C. Song, H. Li, K. Gai, J. Wang, and W. Zhang, "Real-time bidding with multi-agent reinforcement learning in display advertising," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2018, p. 2193–2201.

[26] K. Ren, W. Zhang, K. Chang, Y. Rong, Y. Yu, and J. Wang, "Bidding machine: Learning to bid for directly optimizing profits in display advertising," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 4, pp. 645–659, 2017.

[27] D. P. Bertsekas, "Dynamic programming and optimal control," *Athena Scientific,*, 1995.

[28] ——, "Dynamic programming and stochastic control." 1976.

[29] G. Tesauro and G. Galperin, "On-line policy improvement using monte-carlo search," *Advances in Neural Information Processing Systems*, vol. 9, 1996.

[30] S. Boyd, M. T. Mueller, B. O'Donoghue, Y. Wang *et al.*, "Performance bounds and suboptimal policies for multi–period investment," *Foundations and Trends® in Optimization*, vol. 1, no. 1, pp. 1–72, 2013.

[31] M. W. Ulmer, *Approximate dynamic programming for dynamic vehicle routing*. Springer, 2017, vol. 1.

[32] S. E. Dreyfus, "Dynamic programming and the calculus of variations," *Journal of mathematical analysis and applications*, vol. 1, no. 2, pp. 228–239, 1960.

[33] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.