Advertiser-first: a Receding Horizon Bid Optimization Strategy for Large-scale Online Advertising on E-commerce Platforms

Anonymous Author(s)*

ABSTRACT

Online advertising on e-commerce platforms, which allocates impressions according to consumers' and advertisers' interests, has become an emerging advertising paradigm. We consider solving a large-scale real-time constrained bid optimization problem in online advertising under a dynamic environment. We first model the problem as a Markov decision process, which reveals that the optimal bidding strategy involves a family of functions that depend on remnant budgets and received benefits. To fulfill real-time requirements, a rollout mechanism is introduced to formulate the optimal strategy parametrically, which can be further solved by linear programming with the optimal guarantee. Then an appropriate sampling of historical auction records is introduced to balance online solving consumption and accuracy. Furthermore, to improve stability, we develop the dynamics of the bidding process and reformulate it as a tracking control problem solved by optimal control theory. Finally, a series of numerical experiments on industrial datasets show that it outperforms several works, including state-of-the-art and well-implemented methods in the industry.

CCS CONCEPTS

• **Information systems** → **Computational advertising**; **Sponsored search advertising**; • **Applied computing** → *Online auctions.*

KEYWORDS

Computational advertising, bid optimization, rollout, linear programming, optimal control

ACM Reference Format:

Anonymous Author(s). 2023. Advertiser-first: a Receding Horizon Bid Optimization Strategy for Large-scale Online Advertising on E-commerce Platforms. In *Proceedings of Proceedings of the 2024 International Conference on Management of Data (SIGMOD 2024).* ACM, New York, NY, USA, 14 pages. https://doi.org/X.X

1 INTRODUCTION

With the widespread use of the Internet, online (Internet) advertising has become one of the most popular business fashions since the last decade [5, 5, 13, 28, 42]. Between 2021 and 2022, global online advertising revenues grew 10.8% year-over-year (YoY) totaling $209.7 billion, and overall revenues increased $20.4 billion YoY[9].

SIGMOD 2024, June 11–16, 2024, Santiago, CHILE

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/X.X

1.1 Overview

Online advertising is used in various scenarios, e.g., e-commerce (Amazon, Taobao), searching (Google), etc.[5, 29]. As a substantial industrial cluster, different applications face varying problems. Limited by interests, we focus on e-commerce platforms in this paper, where online advertising helps advertisers to advertise their items.

For each opportunity to display ads in front of consumers, which is called an **impression**, online advertising holds an auction, also called **real-time bidding (RTB)**, for interested advertisers and decide who wins the impression [50–52]. Typically, **e-commerce platforms** provide a convenient, intelligent, and automatic alternative, which aggregates historical information, uses impression-based algorithms, provides quantified objectives, and helps advertisers react individually to each valued impression[32, 49]. **For convenience**, RTB specifically means the one on e-commerce platforms we focus on unless otherwise specified in the rest.

The process of RTB is briefly shown in Fig. 1, where platforms play a vital role in the interaction between advertisers and consumers. The whole process ((a) - (j)) usually accomplishes within 0.1 seconds to avoid long page-loading times at consumer terminals [5, 49, 56]. Advertisers only need to provide advertising requirements to platforms, also called **ad campaigns**, including key performance indicators (KPIs), target groups, constraints, etc. Then **(a)** A consumer searches for something on platforms, triggering an impression. **(b)** Platforms send an ad request to an auction agent implemented on their backend and servers, together with impression features (e.g., the searched words, etc.). Interested advertisers would be called back, and we use advertiser A as an example. **(c)** Evaluate the impression value based on its features, target groups, etc. **(d)** Load marketing objectives (KPIs and constraints) from the ad campaign. **(e)** Load auxiliary, e.g., the historical auction logs, etc. **(f)** The **bid optimization algorithm** optimizes the bidding strategy, which focuses on maximizing (or minimizing) KPIs and holds constraints. **(g)** Advertiser A bids a price based on the optimized bidding strategy. **(h)** All advertisers return bids to the agent. **(i)** Auction agent chooses the winner based on the auction mechanism and returns its ad to platforms. **(j)** The consumer sees the result page where the winner's ad is embedded.

In this paper, we focus on **bid optimization** (Step (g) in Fig.1) and its corresponding problem because it is a core component for helping advertisers bid for each impression and achieve objectives. Solving the bid optimization is **challenging** because it requires

- **(C1) Efficient.** More than a billion impressions emerge on mainstream e-commerce platforms daily, where each impression should be auctioned within 0.1 seconds [13, 24, 55]. It requires the designed bid optimization to be efficient enough to bid a reasonable price within a significantly short frame.
- **(C2) Scalable.** Different advertisers have varying requirements, inducing the designed bid optimization method needs to be scalable for various advertising capabilities.

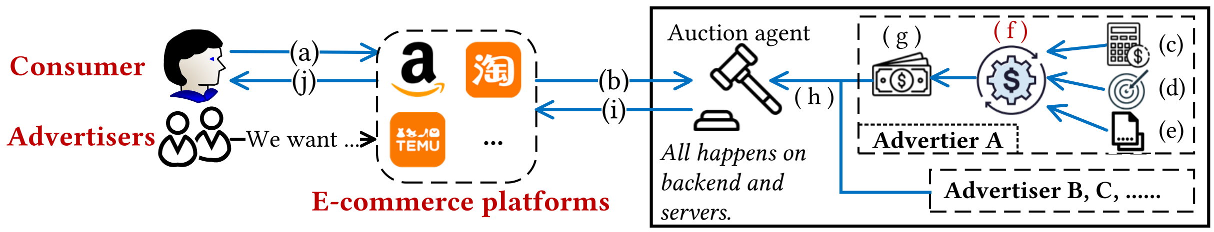

Figure 1: A brief illustration of RTB process. The blue lines are real-time interactions: (a) A consumer searches for something. (b) Send an ad request and call back interested advertisers; A is an example. (c) Evaluate values. (d) Load marketing objectives. (e) Load supplementary. (f) Bid optimization. (g) Bid a price. (h) All advertisers return bids. (i) Send the winner's ad back to platforms. (j) See the page together with the ad. The black line is periodical: advertisers provide advertising requirements.

- **(C3)** **Adaptive and stable.** Since the market fluctuation is inevitable [12, 19], it asks for a stable and adaptive method.

1.2 Preview of Our Work and Contributions

This paper aims to solve the bid optimization problem on e-commerce platforms. As shown in Fig. 2, our methodology contains three parts:

- (a) **The constructed bid formulation.** We constructed a bid formulation as the bidding strategy to overcome **(C1)** and **(C2)**. It has finite optimized parameters, which equals the number of constraints. More importantly, we prove it is one of the optimal strategies.
- (b) **Online parameters optimization.** For the adaptiveness in **C3**, we further design the receding horizon optimization process to adjust parameters in the strategy based on real-time platform feedback on consumption and benefits.
- (c) **Optimal tracking.** Consider the stabilization in **(C3)**, an optimal tracking algorithm is proposed as an auxiliary.

The whole methodology is self-contained. Its inputs are the auctioned impression values, marketing objectives, the sampled auction logs, and platform feedback on current consumption and benefits. Its output is the bid price for the auctioned impression.

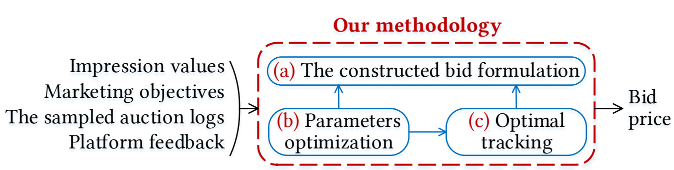

Figure 2: The proposed bid optimization methodology.

Our contributions mainly consist of

- (a) A bid optimization methodology is proposed, combining rollout, linear programming, sampling, and optimal control. The work is based on a real commercial scenario and verified on the industrial dataset.
- (b) To overcome **(C1)** and **(C2)**, we parameterize the optimal bidding strategy with a formula that only has parameters for the number of constraints in the problem. It periodically updates by re-solving a receding horizon optimization to improve adaptiveness in **(C3)**.
- (c) An optimal tracking algorithm is proposed to improve stabilization as mentioned in **(C3)**. To this end, The tracking dynamics are theoretically derived first, and we correct certain misinterpretations introduced in previous works. Finally, we solve it by using the optimal control theory.

The remainder is organized as follows. Section 2 formulates the problem. Section 3 analyzes the structure of bidding functions, which prepares for the constructed bid formulation and parameters optimization in Section 4. Section 5 designs the tracking algorithm. Experiments are shown in Section 6. Finally, the related works are summarized in Section 7, and Section 8 concludes the paper.

2 PROBLEM FORMULATION

2.1 Notations

First, some useful notations are listed in Table 1.

2.2 Problem of Interests

In the bid optimization problem, we aim to find a strategy π that bids a price b for each coming impression with feature $\mathbf{x}$, optimizes the expected KPIs (v), and holds constraints (B and C). Formally,

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

where the bid price $b_t = \pi_t$, CPA constraints may exist k simultaneously that the subscript $j \in \{1,\ldots,k\}$ is used to distinguish them. Indicator function $\mathbf{1}(w(\mathbf{x}_t),\pi_t)$ means the t-th auction result, if $b_t \ge w(\mathbf{x}_t)$, then $\mathbf{1}(w(\mathbf{x}_t),\pi_t) = 1$; otherwise, it equals to 0. As shown in Fig. 1, there exists the value evaluation, which maps $\mathbf{x}$ to v and v_a [31, 51, 53]. It is another vital problem in RTB that exceeds our interests, including influence analysis[23, 37, 38, 57], click-through rate estimation [22, 48, 58], conversion rate estimation [44, 45], landscape prediction [43], feature construction [2, 55], etc. For convenience, we assume it is prior to this paper. Besides, c, w, v, and v_a are related to $\mathbf{x}$ [50, 52, 56]. Thus, we can write $v = v(\mathbf{x})$, $v_{a,j} = v_{a,j}(\mathbf{x})$, $c = c(\mathbf{x})$, and $w = w(\mathbf{x})$.

Table 1: Notations

Notation	Desperation
$\mathbf{x}$	**Impression feature**, $\mathbf{x} \in \mathcal{X}$, a high dimensional vector to describe an impression.
v	**Impression value**, $v \geq 0$, the estimated KPI of an impression if an advertiser wins the auction.
v_a	**Impression action value**, $v_a \geq 0$, the estimated amount of action(s) an impression takes if an advertiser wins the auction, e.g., clicks.
T	**Auction scale**, $T \in \mathbb{Z}_+$, the total number of auctions an ad campaign receives within a specific interval, usually one-day.
b	**Bid price**, $b \geq 0$, the cost that an advertiser wants to pay for the auctioned impression.
w	**Winning price**, $c \geq 0$, the lowest bid price to win its auction for an impression.
c	**Cost**, $c \geq 0$, the expense for the impression that advertisers win.
B	**Budget constraint**, $B \geq 0$, the total cost that an ad campaign can use.
C	**Cost-per-action (CPA) constraint**, $C \geq 0$, the maximum average cost an ad campaign can pay for a specified action (e.g., a click) observed on the delivered impression.
p^{CTR}	**Click-through rate (CTR)**, $p^{\text{CTR}} \in [0,1]$, the estimated probability that consumers click through an ad when they see it.
p^{CVR}	**Conversion rate (CVR)**, $p^{\text{CVR}} \in [0,1]$, the estimated probability that consumers buy the item after clicking through an ad.

Two more settings need to clarify. Firstly, various features can describe an impression, such as cookies, time, etc. Mathematically, $\mathbf{x}$ is 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 [50, 52]. Thus, we can solve (1) in the expected sense. The other is the cost, c, which relates to $\mathbf{x}$, the auction mechanism, and the billing method [50, 52, 55]. In this paper, the **second-price auction (SPA)** is used —— the highest wins and pays for the second-highest price [41]. Meanwhile, the billing method is **click-based** —— advertisers only pay once for a clicked-through ad, which is more acceptable and popular nowadays[19]. In contrast, in most of the previous works, billing has nothing to do with whether an ad has been clicked or not, also called display-based [27, 49, 54, 56].

To sum up, we hope to find a bidding strategy that maximizes the expected total amount of v. Constraints are the budget, which does not exceed B, and the CPA, which limits the average cost of interested actions (e.g., clicks).

3 STRUCTURE OF BIDDING FUNCTIONS

Bid optimization is a typical sequential decision-making problem. A remarkable class of well-studied sequential decision-making problems is the Markov decision process (MDP).

3.1 Markov Decision Process Modelling

In this part, we formulate (1) into an MDP form. Formally, we consider a finite time-horizon MDP $\{\mathcal{S}, \mathcal{B}, p, r, T\}$:

- **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, $s_t = [B_t \ \mathbf{x}_t^\top]^\top \in \mathcal{S}$.
- **Bid price set** $\mathcal{B}$. For the t-th auction, bid price $b_t \in \mathcal{B}$.
- **State transition probability**, $p = \mathbb{P}(s_{t+1}|s_t, b_t)$. For a state B_t and action 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 state transition of $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ is not affected by b_t while is fully described by $p_{\mathbf{x}}$. The above two transition rules together characterize the state transition probability.
- **Reward**, $r = r(s_t, b_t)$ is:

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

  where $\lambda_j > 0$ is the Lagrange multiplier that remakes the objective with CPA constraints as punishment terms. The reason for doing so is whether CPA constraints are violated or not can be tested from real-time accessed data [25, 26, 56].
- **Time horizon**. The auction scale T is the time horizon.

Further, we can write $\pi_t = \pi_t(s_t)$ and solve:

$$\underset{\pi_t, t=1,\dots,T}{\arg\max} \sum_{t=1}^{T} \mathbb{E}\left[r_t\right]. \tag{2}$$

Note that the budget constraint has been encoded in MDP, which does not explicitly appear in (2). In some of the previous studies [24, 27, 39, 49, 52], researchers do not consider the influence of the dynamic consumed budget and thus solve a static optimization problem only with fixed predefined B. These methods can hold optimality but are fragile in the face of market fluctuation. On the other hand, in most learning-based studies [11, 20, 25, 26, 34], budget dynamics have been considered in their MDP models. Though it can achieve the optimal theoretically, enormous resources and data remain bottlenecks. Compared to these two research approaches, we focus on solving the bid optimization problem using MDP modeling together with optimization methods.

Dynamic programming (DP) is usually used to solve such sequential decision-making problems, but it is still **challenging** because:

(a) Since the existence of CPA constraints, (2) is not completely equivalent to the original problem (1).
(b) The problem scale is immense (refer to Section 1), requiring the designed method to be efficient and general enough.
(c) Typically, an ad campaign would receive $0 \sim 10^7$ auctions on mainstream e-commerce platforms daily[24, 25, 49–51]. Using the classical DP to solve (2) is impractical because of the high real-time requirement of bid optimization.

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.

3.2 The Structure of Bidding Functions

Compromise formulas are various approximates of DPs (ADPs) [7, 8, 17, 40], which require a more tractable computation, but ignore in part the availability of extra information. We introduce an approximate method known as the open-loop feedback control (OLFC), which is a rollout mechanism, to solve (2). The core concept of OLFC is to use the currently accessible information as feedback to determine the probability of the future state. However, it calculates the solution as if no further information will be received, using an open-loop optimization over the future evolution of the dynamic[8]. The most crucial characteristic of OLFC is **non-decreasing**, which means the optimized strategy is not worse after each update [8, 17].

Back to (2), we use $\mathbf{x}$ accessed from online as feedback to adjust the current and future bid prices in real-time and ignore the real budget remnant information. Specifically, 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 form:

$$
\begin{aligned}
\arg\max_{\bar{\pi}_t, t=m,\dots,T} \ & \sum_{t=m}^{T} \mathbb{E}\left[v(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))\right] \\
& - \sum_{j=1}^{k} \lambda_j \Bigg(B_m + \sum_{t=m}^{T} \mathbb{E}\left[c(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))\right] \\
& - C_j \left(\sum_{t=m}^{T} C_{j,m} + \mathbb{E}\left[v_{a,j}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t))\right] \right) \Bigg) \\
\text{s.t. } \ & \mathbb{E}\left[\sum_{t=m}^{T} c(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), \bar{\pi}_t(\mathbf{x}_t)) \right] \le R_m.
\end{aligned} \tag{3}
$$

where based on OLFC, $R_m = B - B_m$ is the remnant budget, $C_{j,m} = \sum_{t=1}^{m} v_{a,j}(\mathbf{x}_t)\mathbf{1}(w(\mathbf{x}_t), b_t)$ is the received amount of the j-th interested actions. Thus the altered strategy $\bar{\pi}$ only relates to $\mathbf{x}$. At the same time, (3) with different m and B_m have the same form. At each iteration m, we solve (3) for R_m, then bid a price through $b_m = \bar{\pi}_m(\mathbf{x}_m)$. We further simplify the formulation of (3) via the following operations. First, we consider taking expectations with respect to the randomness of auctions and with respect to $\mathbf{x}_t$'s separately. Given $\mathbf{x}$, we have

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

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 form:

$$
\begin{aligned}
\arg\max_{q_t, t=m,\dots,T} \ & \sum_{t=m}^{T} \mathbb{E}\left[v(\mathbf{x}_t)q_t(\mathbf{x}_t)\right] \\
\text{s.t. } \ & \mathbb{E}[c(\mathbf{x}_t)q_t(\mathbf{x}_t)] \le R_m, \\
& \frac{B_m + \sum_{t=m}^{T} \mathbb{E}[c(\mathbf{x}_t)q_t(\mathbf{x}_t)]}{C_{j,m} + \sum_{t=m}^{T} \mathbb{E}[v_{a,j}(\mathbf{x}_t)q_t(\mathbf{x}_t)]} \le C_j, \ j = 1, \dots, k.
\end{aligned} \tag{4}
$$

Besides, the impression feature $\mathbf{x}$ is i.i.d and draws from the same set $\mathcal{X}$ subject to $p_{\mathbf{x}}$. It indicates that the quantity of the remnant budget B_m affects the strategy of each auction. If we could solve (4), 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 (4) 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 (4).

4 IMPLEMENTATION TRADE-OFF IN SOLVING BIDDING FUNCTION

Though we know the structure of the bidding function that is solved from (4), it is still difficult to solve. The reasons are not only the large-scale and high real-time, but also the difficulties of lacking the exact knowledge about $w(\mathbf{x})$, $p_{\mathbf{x}}$, and other variables in (4).

4.1 Solution Analytics

An auction log is a record of an auctioned impression in the past. As mentioned in Section 3, $\mathbf{x} \in \mathcal{X}$ is i.i.d. and follows $p_{\mathbf{x}}$[24, 25, 49, 52]. Thus, we can approximately identify the variables from the logs; that is, every day's $\mathbf{x}$ recorded in logs can be seen as a collection of samples from $\mathcal{X}$ [24, 49]. Then we replace the expectations in (4) with their values records, generated from logs and solve:

$$
\begin{aligned}
\arg\max_{\hat{q}_t(\hat{\mathbf{x}}_t), t=m,\dots,\hat{T}} \ & \sum_{t=m}^{\hat{T}} \hat{v}_t \hat{q}_t(\hat{\mathbf{x}}_t) \\
\text{s.t. } \ & \sum_{t=m}^{\hat{T}} \hat{c}_t \hat{q}_t(\hat{\mathbf{x}}_t) \le R_m, \\
& \frac{B_m + \sum_{t=m}^{\hat{T}} \hat{c}_t \hat{q}_t(\hat{\mathbf{x}}_t)}{C_{j,m} + \sum_{t=m}^{\hat{T}} \hat{v}_{a,j,t} \hat{q}_t(\hat{\mathbf{x}}_t)} \le C_j, \ j = 1, \dots, k,
\end{aligned} \tag{5}
$$

where the superscript $\wedge$ denotes the value recorded in logs. Besides, the billing method used is click-based (refer to Section 2), which means that $\hat{c}_t = \hat{w}_t \hat{p}_t^{\text{CTR}}$ when we evaluate $\mathbb{E}[c(\mathbf{x}_t)|\mathbf{x}_t]$. Further, the following lemma shows the structure of the solution to (5).

Lemma 4.1. *If (5) is feasible, then there exists an optimal solution $\hat{q}_m^*(\mathbf{x}_m), \dots, \hat{q}_{\hat{T}}(\mathbf{x}_{\hat{T}})$ of (5), which has at least $(\hat{T} - m - k)$ number of 0 and 1, where $m = 1, \dots, \hat{T}$ and $t = m, \dots, \hat{T}$.*

Limited by the space, the proof can be found in the anonymous supplement [4], Section 1. Secondly, $\hat{q}_t(\hat{\mathbf{x}}_t) \in [0, 1]$ implies there exists additional $(\hat{T} - m + 1)$ constraints in (5), that is, $\hat{q}_t(\hat{\mathbf{x}}_t) - 1 \le 0$, $t = m, \dots, \hat{T}$. Then, we can write (5) to its dual form as:

$$
\begin{aligned}
\arg\min_{\alpha, \beta_j, \gamma_t} \ & R_m \alpha + \sum_{t=m}^{\hat{T}} \gamma_t + \sum_{j=1}^{k} \beta_j (B_m - C_{j,m} C_j) \\
\text{s.t. } \ & \hat{c}_t \alpha + \sum_{j=1}^{k} \left(\hat{c}_t - C_j \hat{v}_{a,j,t}\right) \beta_j - \hat{v}_t + \gamma_t \ge 0, t = m, \dots, \hat{T}, \\
& \alpha \ge 0, \beta_j \ge 0, \gamma_t \ge 0, t = m, \dots, \hat{T}, j = 1, \dots, k.
\end{aligned} \tag{6}
$$

Let $\alpha^*, \beta_1^*, \dots, \beta_k^*$ be the optimal to (6) and define

$$
\mu_t = \frac{1}{\hat{p}_t^{\text{CTR}}} \frac{\hat{v}_t + \sum_{j=1}^{k} \beta_j^* C_j \hat{v}_{a,j,t}}{\alpha^* + \sum_{j=1}^{k} \beta_j^*}. \tag{7}
$$

Lemma 4.2. *Suppose (5) is feasible. Consider the optimal solution $\hat{q}_t^*(\mathbf{x}_t)$ studied in Lemma 4.1, $\forall t = m, \dots, \hat{T}$, $m = 1, \dots, \hat{T}$, $\hat{q}_t^*(\hat{\mathbf{x}}_t) = 1$ if $\mu_t > \hat{w}_t$; $\hat{q}_t^*(\hat{\mathbf{x}}_t) = 0$ if $\mu_t < \hat{w}_t$; $\hat{q}_t^*(\hat{\mathbf{x}}_t) \in [0, 1]$ if $\mu_t = \hat{w}_t$.*

The proof refers to the anonymous supplement [4], Section 2, which is mainly based on the dual theory. **Finally, we have**

Theorem 4.3. *Suppose that $\hat{T} \gg k$ and that (5) is feasible. The following bidding rule*

$$\bar{\pi}_t(\hat{\mathbf{x}}_t) = \frac{1}{\hat{p}_t^{\mathrm{CTR}}} \frac{\hat{v}_t + \sum_{j=1}^{k} \beta_j^* C_j \hat{v}_{a,j,t}}{\alpha^* + \sum_{j=1}^{k} \beta_j^*}. \tag{8}$$

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 SPA mechanism.

The proof refers to the anonymous supplement [4], Section 3. **Please note** that $\hat{T} \gg k$ is practical as $\hat{T}$ is around $[10^4, 10^5]$ in most cases [9, 12, 13, 24, 25, 49, 50]. In contrast, the number of CPA constraints is zero or one in reality, $k = \{0, 1\}$ [19, 24, 49, 51, 52]. When $k = 0$, the problem is also called **Budget-Constraint Bidding (BCB)**, $k = 1$ is called **Multi-constraint Bidding (MCB)**. BCB is a special case of MCB. Thus, for a batch of $\hat{T}$ impressions, when we compare auction results of Lemma 4.1 and Theorem 4.3, **the number of different results would not exceed** $k + 1$, which is an extremely small quantity compared to $\hat{T}$. In fact, this is the best we can get because those auctions with different results are right at the critical point that violates constraints.

We use the rule (8) as the bid pricing rule obtained from solving (3) 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 (8), that is,

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

4.2 Sample-based approximate solution

To work out the parameters α and β, we need to solve a dual LP problem (6) in a horizon-receding manner. The simplex method and others [36, 47] can be used to solve, but it is still costly to meet the real-time requirement when $\hat{T}$ is large ($\hat{T} \geq 10^6$).

Inspired by *bootstrap* [15], we propose to use uniformly sampled logs to solve $\bar{\pi}$. The pseudo-code for approximately solving (6) using sampling is reported in Algorithm 1. To understand Algorithm 1, some notations need to be defined. **An auction record** in the log for $\mathbf{x}_t$ is denoted as a tuple ω_t and consists of

$$\omega_t = \{year/month/day/hour/minute/second, \hat{v}, \hat{v}_{a,1}, \ldots, \hat{v}_{a,k}, \hat{w}\},$$

where the string 'year/month/day/hour/minute/second' is the impression occurrence time. All records in a day are archived as an auction log, denoted as $\Omega = \{\omega_1, \ldots, \omega_{\hat{T}}\}$. And the sampling size is regarded as N, the solution solved by the sampled N records is denoted as $\{\alpha_N, \beta_{1,N}, \ldots, \beta_{k,N}\}$. For Algorithm 1, we have

Theorem 4.4. *Suppose (6) is feasible for BCB, and there exists the optimal solution α^*. As the sampling size N increases, the result in Algorithm 1, α_N converges to α^* almost surely.*

The proof refers to the anonymous supplement [4], Section 4. The sketch is that for BCB, solving (6) actually is a quantile estimation problem of a complex random variable, stemming from $\mathbf{x}$. And the sampling process does not affect the quantile [10, 35].

Algorithm 1 Sample-based approximate solution

1: **Input:** Constraints $B, C_1, \ldots, C_k$, auction log Ω, sample size N.
2: **Initialization:**
3: $\quad \hat{T}$ = the number of records in Ω,
4: $\quad t = 1, B_t = 0, C_{1,t} = 0, \ldots, C_{k,t} = 0$.
5: **When the t-th auction arrives:**
6: $\quad$ Delete records in Ω whose 'hour/minute/second' is earlier than the current time.
7: $\quad \hat{T}$ = the number of records in Ω.
8: $\quad R_t = B - B_t$.
9: $\quad$ **If** $\hat{T} \leq N$ **then**
10: $\quad\quad$ Solve $\alpha_N, \beta_{1,N}, \ldots, \beta_{j,N}$ based on (6) with Ω, R_t, $C_{1,t}, \ldots, C_{k,t}, C_{1,t}, \ldots, C_{k,t}$.
11: $\quad$ **Else**
12: $\quad\quad$ Sample a subset Ω_t with N records from Ω.
13: $\quad\quad$ Solve $\alpha_N, \beta_{1,N}, \ldots, \beta_{j,N}$ based on(6) with Ω_t, $NR_t/\hat{T}$, $C_{1,t}, \ldots, C_{k,t}, C_{1,t}, \ldots, C_{k,t}$.
14: $\quad$ **End if**
15: $\quad$ Update (8), and bid a price b_t.
16: $\quad$ Send b_t to the auction agent.
17: $\quad$ Receive the cost c and benefits $v_{a,1}, \ldots, v_{a,k}$.
18: $\quad B_{t+1} = B_t + c, C_{1,t+1} = C_{1,t} + v_{a,1}, \ldots, C_{k,t+1} = C_{k,t} + v_{a,k}$.
19: $\quad t = t + 1$.
20: $\quad$ Wait for the next auction and **go back** to Step 5.

We further guess Theorem 4.4 also is true for MCB (that is, $\forall\ j = 1, \ldots, k,\ \beta_{j,N} \to \beta_j^*$ as $N \to \infty$), but it is tough to elaborate since it relates to multiple random variables, and those are coupled with dual variables. To our best knowledge, no work has answered the convergence guarantee for the sampling in solving BCB or MCB until now. On the other hand, the mainstream fashion still is BCB such that Theorem 4.4 can satisfy most cases, and the extension of MCB is our main work in the future. Finally, Section A proposes a series of experiments for empirically setting a feasible N that balances the accuracy and efficiency in practical use (also for verifying the guess of MCB). **We find that** $N = 10000$ **is one of the acceptable choices, which uses around** 3% **of data and achieves more than** 90% **accuracy.**

5 LINEAR-QUADRATIC-REGULATION TRACKING MECHANISM

In this section, we focus on the stabilization of the bid optimization method in reality because:

(a) Consumers' actions after seeing ads usually are time-delayed (e.g., click or not). For an ad campaign, the time is much longer than the interval between two consecutive auctions.
(b) Algorithm 1 highly depends on the consumed budget and others, which come from the billing module. As set in Section 3.2, the click-based billing method implies that the billing module also runs periodically [24, 51].
(c) Millions of ad campaigns are implemented on e-commerce platforms[24, 25, 49, 51, 55], sampling new subsets simultaneously may be stressful for infrastructure.
(d) Market fluctuation is inevitable [49, 50, 54].

For the first two challenges, platforms use a billing module to periodically (e.g., every 5 minutes) count each ad campaign's cost (and benefits) and report forms. Consequently, it makes the bid optimization run periodically, widely adopted in industry and previous works[11, 24, 25, 34, 42, 46, 47, 49, 51, 54]. For the last two, a compromise is to use a double-cycle (optimization-tracking) structure [27, 49, 52, 54]. Specifically, as shown in Fig.3,

- **Optimization.** Suppose Algorithm 1 runs first at timestamp t_a, then the next timestamp it runs is $t_a + \tau_a$, where τ_a is greater than a feasible time that all ad campaigns can finish sampling the new subset. When Algorithm 1 runs at t_a, it also outputs referred trajectories for the time remaining up to $t_a + \tau$, which contain the consumed budget and the received amount of actions.
- **Tracking.** Tracking runs multi-time in $[t_a, t_a + \tau_a)$. Its running time interval τ_t is greater than the feedback time of billing modules. Due to market fluctuation, it adjusts parameters in the bid formulation such that the current consumed budget and the received amount of actions can track their references proposed by Algorithm 1, respectively.

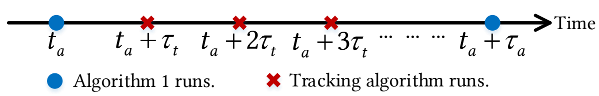

Figure 3: Optimization-tracking double-cycle structure.

Fundamentally, tracking is a stopgap when the platform lacks sufficient resources to address the constructed bid optimization problem before the latest feedback expires. If we can run Algorithm 1 quickly enough to meet the latest feedback from billing modules or use an end-to-end machine-learning-based method (e.g.,[11, 20, 25, 26, 34], etc.), the tracking algorithm can be omitted. On the other hand, if we set a long update interval, market fluctuations may destabilize the bidding process in this interval [49, 54], which requires a tracking algorithm to improve stabilization until the strategy has been updated.

Compared to the previous tracking algorithms [49, 52, 54, 56], we first model the tracking dynamics from the RTB process such that the designed algorithm can maintain the optimal tracking performance based on it. Thus, our algorithm is model-based with a theoretical optimality guarantee, while the previous ones are data-based, which is empirical.

5.1 Tracking problem formulation

The derivation of the dynamics with k CPA constraints is lengthy and tedious, with only one CPA constraint at most in most cases, and cost-per-click (CPC) is the most used CPA constraint [19, 24, 49, 52]. **Therefore, we derive dynamics with budget and CPC constraints in this section as an example.** Moreover, the entire process is universal, and we would point out the discrepancy for other CPA constraint settings in brackets.

The CPC constraint is defined as the average click cost that does not exceed a predefined C. In this sense, the impression action value $v_a = p^{\mathrm{CTR}}(\mathbf{x}_t)$, and based on Theorem 4.3,

$$b(\mathbf{x}_t) = \pi_t(\mathbf{x}_t) = \frac{v(\mathbf{x}_t) + \beta C p^{\mathrm{CTR}}(\mathbf{x}_t)}{(\alpha+\beta)p^{\mathrm{CTR}}(\mathbf{x}_t)}. \tag{10}$$

An intuitive idea is to track references of the consumed budget and instant CPC at each timestamp. Indeed, in [27, 49, 54] and related studies, the authors designed state feedback controllers to track them, although they have some misinterpretations. In the control theory [16], state feedback can directly reflect the internal mode of a system. However, CPC (CPA) is not the state. Instead, it is the quotient of the consumed budget and the total amount of clicks (the total amount of received actions). In other words, CPC (CPA) is the dynamics' output and should be regulated by output feedback controllers, which may be nonlinear.

5.1.1 *Dynamics models.*

Therefore, the dynamics states are **the consumed budget and the total amount of clicks**. As shown in Fig.3, the first timestamp that Algorithm 1 runs is t_a. In the time interval $[t_a, t_a + \tau_a)$, suppose the tracking algorithm runs n times, we have the following dynamics,

$$\begin{bmatrix} B_{t_{c,m+1}} \\ L_{t_{c,m+1}} \end{bmatrix} = \begin{bmatrix} B_{t_{c,m}} \\ L_{t_{c,m}} \end{bmatrix} + \begin{bmatrix} \bar{c}_{t_{c,m}} \bar{p}^{\mathrm{CTR}}_{t_{c,m}} \mathbb{E}[n_{t_{c,m}} | \alpha_{t_{c,m}}, \beta_{t_{c,m}}] \\ \bar{p}^{\mathrm{CTR}}_{t_{c,m}} \mathbb{E}[n_{t_{c,m}} | \alpha_{t_{c,m}}, \beta_{t_{c,m}}] \end{bmatrix} + \begin{bmatrix} \Delta B_{t_{c,m}} \\ \Delta L_{t_{c,m}} \end{bmatrix}, \tag{11}$$

where $t_{c,m}$ is the timestamps of the m-th tracking algorithm runs, $m = \{1, \dots, n\}$; $L_{t_{c,m}}$ is the total amount of clicks until $t_{c,m}$; $\alpha_{t_{c,m}}$, $\beta_{t_{c,m}}$ are parameters in (10) at $t_{c,m}$. Market fluctuations are ΔB and ΔL, which can be regarded as Gaussian disturbanes [12, 13, 50, 52]. In $[t_{c,m}, t_{c,m+1})$, assuming there are $n_{t_{c,m}}$ auctions, $\mathbb{E}[n_{t_{c,m}} | \alpha_{t_{c,m}}, \beta_{t_{c,m}}]$ is the expected total number of winning auctions with bid parameters $\{\alpha_{t_{c,m}}, \beta_{t_{c,m}}\}$. For those $n_{t_{c,m}}$ impressions, their expected average cost and the amount of clicks are $\bar{c}_{t_{c,m}}$, $\bar{p}^{\mathrm{CTR}}_{t_{c,m}}$, respectively. Variables $n_{t_{c,m}}$, $\bar{c}_{t_{c,m}}$, $\bar{p}^{\mathrm{CTR}}_{t_{c,m}}$ can be known by auction logs.

5.1.2 *Tracking performance metric.*

For each $t_{c,m}$, our objective is to design a tracking algorithm to adjust $\alpha_{t_{c,m}}$ and $\beta_{t_{c,m}}$ such that the current states follows the referred $B^r_{t_{c,m}}$ and $L^r_{t_{c,m}}$. Consequently, we can define the following squared error metric J_{t_a} to measure the tracking performance during the time interval $[t_a, t_a + \tau_a)$,

$$\begin{aligned} J_{t_a} = & \sum_{m=1}^{n+1} \left[\frac{\theta}{B^2} \left(B_{t_{c,m}} - B^r_{t_{c,m}} \right)^2 + \frac{1-\theta}{L^2} \left(L_{t_{c,m}} - L^r_{t_{c,m}} \right)^2 \right] \\ & + \sum_{m=1}^{n} \left[\frac{\eta_\alpha}{\alpha^2_{t_a}} \left(\alpha_{t_{c,m}} - \alpha_{t_a} \right)^2 + \frac{\eta_\beta}{\beta^2_{t_a}} \left(\beta_{t_{c,m}} - \beta_{t_a} \right)^2 \right], \end{aligned} \tag{12}$$

where α_{t_a}, β_{t_a} are basis in (10) proposed by Algorithm 1 at t_a. Normalized factor $L = B/C$. For each $t_{c,m}$, the reference is $L^r_{t_{c,m}} = B_{t_{c,m}}/C$. Scalar $\theta \in [0, 1]$ measures the interested tracking of states. Non-negative scalars η_α, η_β are smooth weights of control signals to avoid aggressive adjustments. (The above is universal for other CPA constraints by replacing the related variables and definitions.)

5.1.3 *Derivation of* $\mathbb{E}[n_{t_{c,m}} | \alpha_{t_{c,m}}, \beta_{t_{c,m}}]$.

Since $\mathbf{x}$ being an integrable random variable, p^{CTR} and c are also random variables with

PDFs $f_1(p^{\text{CTR}})$ and $f_2(c)$ (They are regarded as Beta/Gamma distributions [43, 50]), respectively. Then,

$$\begin{aligned}\tilde{n}_{t_{c,m}} &:= \mathbb{E}(n_{t_{c,m}}|\alpha_{t_{c,m}}, \beta_{t_{c,m}}) \\ &= n_{t_{c,m}}\mathbb{P}\left(\alpha_{t_{c,m}} p^{\text{CVR}} + \beta_{t_{c,m}} C - c \geq 0\right) \\ &= n_{t_{c,m}}\left[1 - \int_0^{c_{\max}} F_1\left(\frac{c - \beta_{t_{c,m}} C}{\alpha_{t_{c,m}}}\right) f_2(c)\mathrm{d}c\right],\end{aligned}$$

where $c_{\max}$ is the maximum cost for an impression set by the platform, F_1 is the cumulative density function (CDF) of p^{CTR}.

5.1.4 *Linearization.*

In (11), the states are nonlinear to control signals $\alpha_{t_{c,m}}$ and $\beta_{t_{c,m}}$. A general approach is linearizing around setpoints, which we choose from α_{t_a} and β_{t_a} as follows:

$$\mathbf{z}_{t_{c,m+1}} \approx \mathbf{z}_{t_{c,m}} + \mathbf{G}_{t_{c,m}}\mathbf{u}_{t_{c,m}} + \Delta\mathbf{z}_{t_{c,m}} + \Delta_{t_{c,m}}, \tag{13}$$

where

$$\mathbf{z}_{t_{c,m}} = \begin{bmatrix} B_{t_{c,m}} \\ L_{t_{c,m}} \end{bmatrix},\ \mathbf{u}_{t_{c,m}} = \begin{bmatrix} \alpha_{t_{c,m}} - \alpha_{t_a} \\ \beta_{t_{c,m}} - \beta_{t_a} \end{bmatrix},\ \Delta\mathbf{z}_{t_{c,m}} = \begin{bmatrix} \bar{c}_{t_{c,m}} \bar{p}^{\text{CTR}}_{t_{c,m}} \tilde{n}_{t_a} \\ \bar{p}^{\text{CTR}}_{t_{c,m}} \tilde{n}_{t_a} \end{bmatrix},$$

$$\mathbf{G}_{t_{c,m}} = \begin{bmatrix} \bar{c}_{t_{c,m}} \bar{p}^{\text{CTR}}_{t_{c,m}} & \bar{c}_{t_{c,m}} \bar{p}^{\text{CTR}}_{t_{c,m}} \\ \bar{p}^{\text{CTR}}_{t_{c,m}} & \bar{p}^{\text{CTR}}_{t_{c,m}}, \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{n}_{t_{c,m}}}{\partial \alpha_{t_a}} \\ \frac{\partial \tilde{n}_{t_{c,m}}}{\partial \beta_{t_a}} \end{bmatrix},\ \Delta_{t_{c,m}} = \begin{bmatrix} \Delta B_{t_{c,m}} \\ \Delta C_{t_{c,m}} \end{bmatrix},$$

with $\tilde{n}_{t_a} = \mathbb{E}(n_{t_{c,m}}|\alpha_{t_a}, \beta_{t_a})$,

$$\begin{aligned}\frac{\partial \tilde{n}_{t_{c,m}}}{\partial \alpha_{t_a}} &= -\tilde{n}_{t_a}\int_0^{c_{\max}} \frac{\beta_{t_a} C - c}{\alpha^2_{t_a}} f_1\left(\frac{c - \beta_{t_a} C}{\alpha_{t_a}}\right) f_2(c)\mathrm{d}c, \\ \frac{\partial \tilde{n}_{t_{c,m}}}{\partial \beta_{t_a}} &= \frac{C\tilde{n}_{t_a}}{\hat{\alpha}}\int_0^{c_{\max}} f_1\left(\frac{c - \beta_{t_a} C}{\alpha_{t_a}}\right) f_2(c)\mathrm{d}c.\end{aligned}$$

5.1.5 *Tracking error dynamics.*

Comparing the current state to the referred ones, the tracking error dynamics read as follows:

$$\mathbf{e}_{t_{c,m+1}} = \mathbf{e}_{t_{c,m}} + \mathbf{G}_{t_{c,m}}\mathbf{u}_{t_{c,m}} + \mathbf{g}_{t_{c,m}} + \Delta_{t_{c,m}}, \tag{14}$$

where

$$\mathbf{e}_{t_{c,m}} = \begin{bmatrix} B_{t_{c,m}} - B^r_{t_{c,m}} \\ L_{t_{c,m}} - L^r_{t_{c,m}} \end{bmatrix},\ \mathbf{g}_{t_{c,m}} = \begin{bmatrix} B^r_{t_{c,m}} + \bar{c}_{t_{c,m}} \bar{p}^{\text{CTR}}_{t_{c,m}} \tilde{n}_{t_a} - B^r_{t_{c,m+1}} \\ L^r_{t_{c,m}} + \bar{p}^{\text{CTR}}_{t_{c,m}} \tilde{n}_{t_a} - L^r_{t_{c,m+1}} \end{bmatrix}.$$

Consequently, the squared tracking error (12) equals

$$J_{t_a} = \mathbf{e}^\top_{t_{c,n+1}} \mathbf{Q} \mathbf{e}_{t_{c,n+1}} + \sum_{m=1}^{n} \mathbf{e}^\top_{t_{c,m}} \mathbf{Q} \mathbf{e}_{t_{c,m}} + \mathbf{u}^\top_{t_{c,m}} \mathbf{R} \mathbf{u}_{t_{c,m}}, \tag{15}$$

where $\mathbf{Q} = \begin{bmatrix} \frac{\theta}{B^2} & 0 \\ 0 & \frac{1-\theta}{L^2} \end{bmatrix}$, $\mathbf{R} = \begin{bmatrix} \frac{\eta_\alpha}{\alpha^2_{t_a}} & 0 \\ 0 & \frac{\eta_\beta}{\beta^2_{t_a}} \end{bmatrix}$.

5.2 Linear-quadratic-regulation-based tracking

Up to now, we hope to design a tracking algorithm that minimizes J_{t_a}, a typical linear-quadratic-regulation (LQR) control problem. And the optimal controller $\mathbf{u}$ is

$$\mathbf{u}_{t_{c,m}} = -\mathbf{H}^{-1}_{t_{c,m}} \mathbf{G}_{t_{c,m}} \mathbf{P}_{t_{c,m+1}} \left(\mathbf{e}_{t_{c,m}} + \mathbf{g}_{t_{c,m}}\right) - \frac{1}{2}\mathbf{H}^{-1}_{t_{c,m}} \mathbf{G}^\top_{t_{c,m}} \mathbf{q}_{t_{c,m+1}}, \tag{16}$$

where

$$\begin{cases} \mathbf{H}_{t_{c,m}} = \mathbf{R} + \mathbf{G}^\top_{t_{c,m}} \mathbf{P}_{t_{c,m+1}} \mathbf{G}_{t_{c,m}}, \\ \mathbf{M}_{t_{c,m}} = \mathbf{P}_{t_{c,m+1}} - \mathbf{P}_{t_{c,m+1}} \mathbf{G}_{t_{c,m}} \mathbf{H}^{-1}_{t_{c,m}} \mathbf{G}^\top_{t_{c,m}} \mathbf{P}_{t_{c,m+1}}, \\ \mathbf{P}_{t_{c,m}} = \mathbf{Q} + \mathbf{P}_{t_{c,m+1}} - \mathbf{P}^\top_{t_{c,m+1}} \mathbf{G}_{t_{c,m}} \mathbf{H}^{-1}_{t_{c,m}} \mathbf{G}^\top_{t_{c,m}} \mathbf{P}_{t_{c,m+1}}, \\ \mathbf{q}_{t_{c,m}} = 2\mathbf{M}_{t_{c,m}} \mathbf{g}_{t_{c,m}} + \mathbf{M}_{t_{c,m+1}} \mathbf{P}^{-1}_{t_{c,m+1}} \mathbf{q}_{t_{c,m+1}}, \end{cases} \tag{17}$$

with the terminal condition $\mathbf{P}_{t_{c,n+1}} = \mathbf{Q}$, $\mathbf{q}^\top_{t_{c,n+1}} = [0\ 0]$. The above is a universal result for LQR [16], and we do not repeat it here. **To sum up**, the pseudo-code of the LQR-based optimal tracking is presented in Algorithm 2.

Algorithm 2 Linear-quadratic-regulation-based tracking

Input: Constraints B, C, auction log Ω, running times n, the consumed budget reference $\{B^r_{t_{c,1}}, \ldots, B^r_{t_{c,n}}\}$, parameters θ, η_α, η_β, α_{t_a}, β_{t_a} in (12).
Initialization:
    $L = B/C$.
    $\mathbf{Q} = \begin{bmatrix} \frac{\theta}{B^2} & 0 \\ 0 & \frac{1-\theta}{L^2} \end{bmatrix}$,$\mathbf{R} = \begin{bmatrix} \frac{\eta_\alpha}{\alpha^2_{t_a}} & 0 \\ 0 & \frac{\eta_\beta}{\beta^2_{t_a}} \end{bmatrix}$,$\mathbf{P}_{t_{c,n+1}} = \mathbf{Q}$,$\mathbf{q}_{t_{c,n+1}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
    For $m = n$ **to** 1 with step -1:
        Solve (17), and cache $\mathbf{H}_{t_{c,m}}, \mathbf{P}_{t_{c,m}}, \mathbf{q}_{t_{c,m}}$.
    End
    m = 1.
For the m**-th tracking update:**
    Read $B_{t_{c,m}}$ and $L_{t_{c,m}}$ from the billing module.
    $L^r_{t_{c,m}} = B_{t_{c,m}}/C$, $\mathbf{e}_{t_{c,m}} = \begin{bmatrix} B_{t_{c,m}} - B^r_{t_{c,m}} \\ L_{t_{c,m}} - L^r_{t_{c,m}} \end{bmatrix}$.
    Solve $\mathbf{u}_{t_{c,m}}$ based on (16).
    Output: $\begin{bmatrix} \alpha_{t_{c,m}} \\ \beta_{t_{c,m}} \end{bmatrix} = \begin{bmatrix} \alpha_{t_a} \\ \beta_{t_a} \end{bmatrix} + \mathbf{u}_{t_{c,m}}$
    $m = m + 1$.
end

---

6 BIDDING PERFORMANCE EVALUATION

6.1 Experiment settings

6.1.1 *Experiment scenario.*

Following the business service provided on Taobao, the objective is maximizing the expected **Gross merchandise volume (GMV)**. Constraints are **budget** B and **cost-per-click (CPC)** C. Thus, for the bid optimization problem (1), $v(\mathbf{x}) = p^{\text{CTR}}(\mathbf{x})p^{\text{CVR}}(\mathbf{x})i_p$, $k = 1$, $v_a(\mathbf{x}) = p^{\text{CTR}}(\mathbf{x})$, where i_p is the item price. Based on Theorem 4.3, the bidding strategy $b(\mathbf{x}_t) = (p^{\text{CVR}}(\mathbf{x}_t)i_p + \beta C)/(\alpha + \beta)$.

6.1.2 *Dataset.*

The anonymous dataset is provided by Taobao, containing 1000 randomly selected ad campaigns and their auction records. Records are from 06/20/2022 - 06/26/2022 and 03/13/2023 - 03/19/2023, totaling around 600 million records and over 11 GB. Each record consists of

$$\{year/month/day/hour/minute/second, \hat{p}^{\text{CTR}}, \hat{p}^{\text{CVR}}, i_p, \hat{w}\},$$

where the string 'year/month/day/hour/minute/second' is the impression occurrence time. For an auctioned impression, $\hat{w}$ is the winning price, i_p is the item price, $\hat{p}^{\text{CTR}}$ and $\hat{p}^{\text{CVR}}$ are the estimated click-through rate and conversion rate, respectively.

6.1.3 Experiment pipeline. The pipeline follows the RTB process in Fig. 1, but different methods do not compete with each other. The experiment is based on the static environment assumption, which assumes the winning price is fixed for each impression [11, 25, 26, 34, 46, 49, 52]. Each method experiments with the same batch of records, which are sequentially inputted based on their timestamps. To rule out the effects of value estimation of $\mathbf{x}$, we use the records of the estimated values $v = \hat{p}^{\text{CTR}}\hat{p}^{\text{CVR}}i_p$ and $v_a = \hat{p}^{\text{CTR}}$ rather than $\mathbf{x}$, which can be directly used for bidding. When the method bids a price b, an auction agent compares it with the recorded winning price $\hat{w}$ based on SPA. Finally, the auction result (win/lose) is sent to billing modules to calculate the expected consumed budget and the expected amount of clicks.

6.1.4 Experiment pipeline settings.

- When an ad campaign wears out its budget, the auction agent ignores its bid price.
- Billing modules report consumption and benefit every 5 minutes (similar to the platform setting). Thus, each method updates every 5 minutes at most.
- All experiments are implemented on Dataworks, an end-to-end big data development and governance platform provided by Alibaba Cloud[14]. The solver is MindOpt[1].

6.1.5 Compared methods.

(a) **Logs replay**, denoted as **LR**. Each ad campaign uses its bid price recorded in the log, which is based on [21] and has been well-implemented on Taobao for several years.
(b) **A unified solution to constrained bidding**, an end-to-end reinforcement learning method [25, 46], denoted as **USCB**. Various works are learning-based; we choose **USCB** because, as a state-of-the-art, it has been reported to be implemented and work well in the industry. It uses (9) as the bidding strategy. But its network inputs are the normalized remnant budget ratio, the budget spending speed, the left time ratio, and the normalized current CPC. Outputs are parameters in (9).
(c) **Proportional–Integral–Derivative controller** [49, 54], denoted as **PID**. As a state-of-the-art, it is adopted in the industry as a classical control-based method. It also uses (9) as the bidding strategy. Its input is the entire auction record and tracking errors of the consumed budget and CPC. Outputs are parameters in (9).
(d) **Waterlevel-based controller**[52], denoted as **WB**. As one of the earliest optimization-based methods, its ideas are absorbed and further developed in the follow-up work. It solves the functional optimization version of (1) as the strategy. Its inputs are the estimated stochastic models of $\hat{p}^{CTR}$, $\hat{p}^{CVR}$, and $\hat{w}$ in records, and tracking errors of remnant budget and CPC. Output is the bid price.
(e) **Our method** , denoted as **RB**.

Someone interested can refer to Section 7 for more things about optimization-based, learning-based, and control-based methods.

6.1.6 Compared methods settings.

- **USCB**: The network is trained on records from 05/19/2022 - 06/19/2022. Besides, the network is re-trained by results after each experiment. For example, the network would be re-trained by records and results of 06/20/2022 before it experiments with 06/21/2022 records.
- **RB**: The method implementation follows Fig.2, where (a) is the formulation $b(\mathbf{x}_t) = (p^{\text{CVR}}(\mathbf{x}_t)i_p + \beta C)/(\alpha + \beta)$; (b) is Algorithm 1; (c) is Algorithm 2. Specifically, Algorithm 1 runs every $\tau_a = 15$ minutes and uses the previous day's auction logs for optimization with $N = 10000$. Algorithm 2 runs every $\tau_t = 5$ minutes; $\theta = 0.5$, $\eta_\alpha = \eta_\beta = 1$; $\bar{c}$ and the others are based on records from 05/19/2022 - 06/19/2022.
- **PID** and **WB**: Both are solved by records from 05/19/2022 - 06/19/2022, and the previous day's auction log.

6.1.7 Advertising evaluation metrics.

(a) **The normalized expected GMV**, denoted as $\tilde{G}$. It is the maximized objective in experiments that measure advertising performance. **LR** is used as the benchmark against which other methods' results are normalized to rule out the influence of different ad campaign scales.
(b) **The normalized expected cost-per-transaction**, denoted as **CPT**. It tells advertisers how much they pay for a transaction on average and is one of the business metrics to measure advertising capability [19, 24, 25, 50, 55]. **CPT** is the lower, the better. Also, **LR** is used as the benchmark.

6.2 Advertising Performance Comparison

First, we compare different methods' advertising performance. **For the metric** $\tilde{G}$, the results are shown in Fig. 4. To sum up,

(a) In Fig. 4, the part on the left side of the dashed line shows the result from 06/20/2022-06/26/2022; the first seven subfigures show the daily performance by the probability density function (PDF), where the horizontal axis is limited to $[0.5, 1.5]$ for clarity. The eighth subfigure is the seven-day PDF. The rest two bar charts are the average (Avg.) and the standard deviation (Std.). The part on the right side comes from 03/13/2023-03/19/2023, following the same form.
(b) Although the dataset comprises two time periods, nine months apart, **each method works well compared to LR**. Overall, the market is relatively stable because Avg. is close for each method in two periods.
(c) **RB outperforms the compared methods.** First, we can find that while the daily $\tilde{G}$ has an upper bound around 1.5 for all methods, their distributions are not similar. **RB** peaks at about 1.1, further to the right than the other methods. This implies that **RB** achieves higher average performance on $\tilde{G}$, which is also verified in the seven-day PDF and the average of $\tilde{G}$. On the other hand, **RB** is more stable than others because its PDF is more concentrated around the peak, and statistically, its Std. is the lowest.
(d) **RB is more effective and adaptive.** Though all methods' average $\tilde{G}$ is greater than 1, a part of campaigns still exists whose $\tilde{G}$ is lower than 1, which implies the compared method performs weaker than **LR**. It mainly has two reasons: one is that **LR** also is a competitive and well-designed method; the second is more important, almost all methods (whether the compared or others mentioned in Section

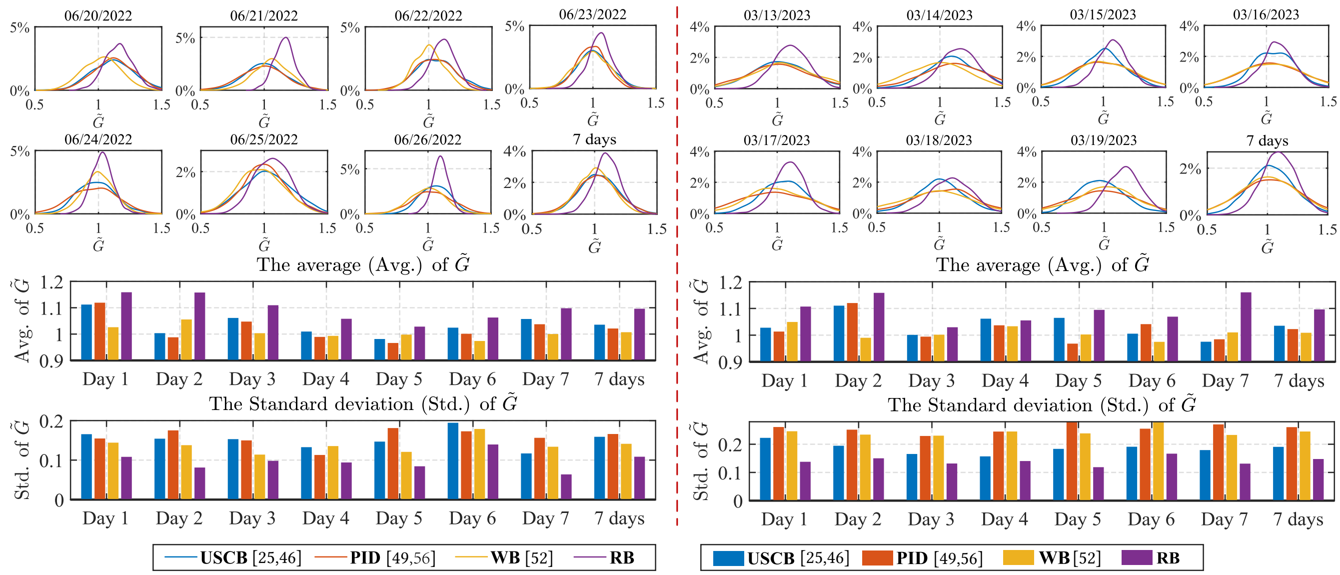

Figure 4: Advertising performance comparisons of $\tilde{G}$. The part on the left side of the dashed red line shows the results from 06/20/2022 - 06/26/2022. The part on the right side is from 03/13/2023 - 03/19/2023. For both parts, the first row presents the PDF for each day; the second row contains the seven-day PDF, the average (Avg.) of $\tilde{G}$, and the standard deviation (Std.) of $\tilde{G}$.

7) assume an ad campaign's impressions follows a stable random process. Thus we can numerically solve the policy based on the history. Unfortunately, for some reasons, e.g., too few records, market fluctuations, etc., it is hard to model these processes with finite records precisely, making part of campaigns uncertain and tricky. Consequently, an ideal method should be adaptive for these cases to obtain more efficient and adaptive performance. In this sense, our method, **RB**, is better than others because no matter the daily PDF or the seven-day PDF, the part of $\tilde{G} < 1$ is significantly smaller. In addition, we also counted the average ratio of $\tilde{G} < 1$ campaigns to all campaigns within seven days, as shown in Table 2. It is clear that although their average $\tilde{G} > 1$, **PID** and **WB** actually perform equally to **LR** because they have around 50% to be weaker or stronger than it. By contrast, **USCB** outperforms **LR** for around 60% campaigns, whereas **RB** is around 70%.

(e) **RB is more stable.** As shown in the fourth row of Fig. 4, **RB** achieves the lowest Std, showing it is the most stable when facing market fluctuations. Further, we compare the results between 2022 and 2023, Std. for **USCB/PID/WB/RB** increases from 0.1589 / 0.1660 / 0.1412 / 0.1085 to 0.1899 / 0.2602 / 0.2431 / 0.1475. This phenomenon is because the supplementary of each method (Specifically, **USCB**'s basic training dataset, **PID/WB**'s random variables' models, **RB**'s parameters in tracking) used to bid optimization comes from 05/19/2022-06/19/2022, whose time is far from the second period (03/13/2023-03/19/2023) in the experiment dataset. This difference can be regarded as another more powerful market fluctuation to test method stabilization. Therefore, even if the market is relatively stable, some fine-grained changes may destabilize methods and enlarge Std.. In this sense, **PID** and **WB** are not stable enough because their absolute increments of Std. are nearly 0.1 (relative increments are over 50%). **USCB** and **RB** have a close absolute increment for Std., but **RB**'s Std. is still significantly lower than **USCB**'s (**USCB** achieves the lowest relative increment due to its basis is bigger than **RB**).

Table 2: The average ratio of $\tilde{G} < 1$ campaigns to all campaigns within seven days.

	USCB[*]	**PID**[*]	**WB**[*]	**RB**
06/20/2022-06/26/2022	0.3703	0.4776	0.4959	**0.2779**
03/13/2023-03/19/2023	0.4299	0.4919	0.5321	**0.3193**

[*] **USCB** [25, 46]. **PID** [49, 56]. **WB** [52].

For the metric CPT, the results are shown in Fig. 5. To sum up,

(a) Fig. 5 follows the same presentation as Fig. 4.
(b) Although the maximized objective is the expected GMV, **advertising capability is also crucial for advertisers**. For example, for the same item, A spends 100 advertising budget to facilitate 10 transactions, the expected GMV is 1000; B spends 50 to facilitate 10 transactions, and the expected GMV is 500. A is better if we only compare the expected GMV; however, the advertising capability, measured by **CPT** (the spent advertising budget divided by the number of transactions), shows that B is better because B's **CPT** is half of A's. In other words, on average, B saves more

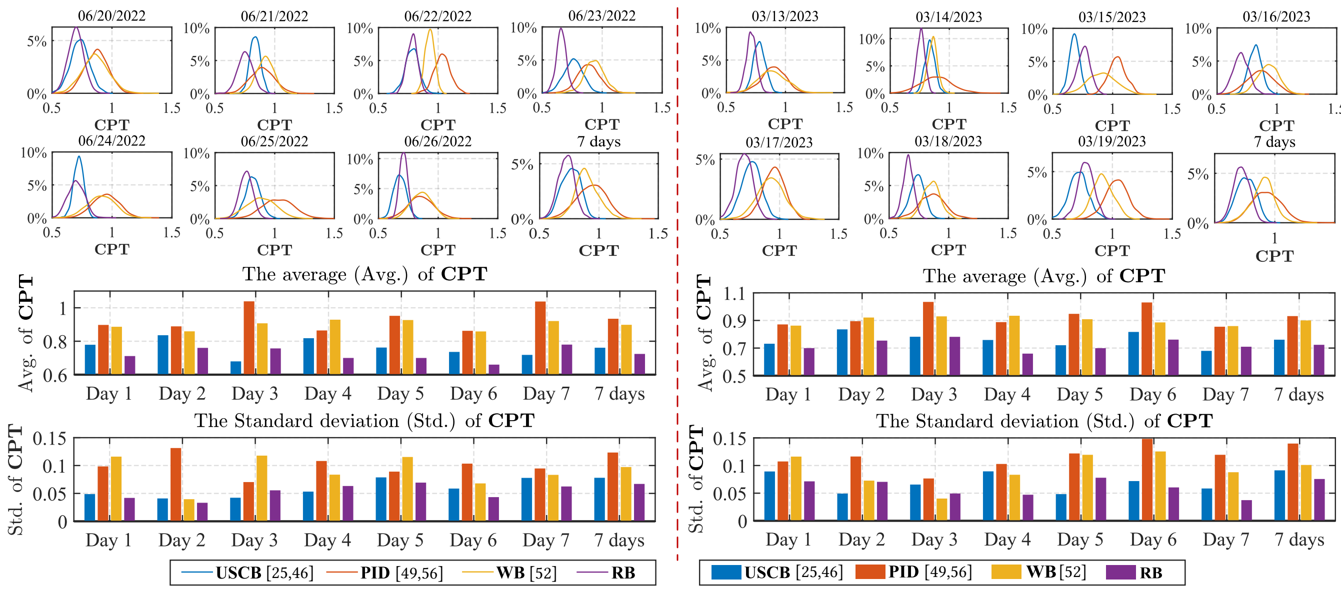

Figure 5: Advertising performance comparisons of CPT. The part on the left side of the dashed red line shows the results from 06/20/2022 - 06/26/2022. The part on the right side is from 03/13/2023 - 03/19/2023. For both parts, the first row presents the PDF for each day; the second row contains the seven-day PDF, the average (Avg.) of CPT, and the standard deviation (Std.) of CPT.

advertising budget for each transaction, which usually is regarded as a cost by advertisers.

(c) **RB has a better advertising capability.** Firstly, according to the daily and the seven-day PDF, **USCB**'s and **RB**'s advertising capability is significantly better than the rest because almost all of their **CPT** is lower than 1. **RB** outperforms **USCB** in 10 of 14 days by comparing their Avg. and Std.. On the other hand, for 06/20/2022-06/26/2022, the seven-day Avg/Std of **USCB** is 0.7612/0.0781, while **RB** is 0.7245/0.0681; for 03/16/2023-03/19/2023, the seven-day Avg/Std of **USCB** is 0.7811/0.0923, while **RB** is 0.7351/0.0727. All of these imply although **USCB**'s advertising capability is competitive and powerful, **RB** performs to the next level.

6.3 Constraint Maintenance

Except for advertising performance, constraint maintenance is also vital. No matter whether in reality or the experiment, budget is a 'hard' constraint: an ad campaign would be locked if its budget is exhausted. By contrast, CPC (generally, CPA) is a 'soft' constraint [19, 24, 25, 50], which can be broken within an acceptable range because CPC (CPA) is not as intuitive as budget to set. Advertisers' settings may not be reasonable and feasible for solving the bid optimization problem. Therefore, we count the CPC constraint violation ratio in Table 3. In summary,

(a) **LR never violates the CPC constraint because** it uses a "hard" threshold, whose bids are truncated by the constraint. As a method used maturely in the industry for several years, this operation may not be elegant, but it definitely works for constraint maintenance. Please note that CPC (generally CPA) is a metric to measure the average unit cost during the day. Therefore, to optimize the objective, we can bid a relatively high price for those high-value impressions, making the current CPC (CPA) unsatisfactory in a short period, but as long as it is satisfied at the end of the day. However, **LR** strictly limits bid prices' upper bound, which may induce advertisers to miss out on high-value impressions that would break the CPC constraint.

(b) **USCB, PID, WB, and RB** all potentially violate the CPC constraint since they do not have such a 'hard' threshold as **LR**. Meanwhile, it readily concludes that **RB achieves the lowest CPC constraint violation ratio**.

(c) While the CPC constraint can be violated within a range, advertisers would be disappointed if we exceeded too much. To measure the degree of those violating campaigns, define

$$V_C = C_{\text{ad}}/C - 1, \tag{18}$$

where C_{ad} is the actual CPC of an ad campaign in experiments, we hope V_C is as close to 0 as possible. Then we compare their average and maximum in Table 4. Compared to **PID/WB**, it is no doubt that **USCB** and **RB** decrease the average of V_C, and their performances are close. However, their worst cases, the maximum of V_C, have remarkable gaps. **PID**'s maximal V_C is nearly 0.9, implying an ad campaign's actual CPC is almost 1.9 times its constraint. Though it may partly be because the setting is too small or the market changes a lot, it reflects the method lacks stabilization when facing tricky. In contrast, **RB works more competitively since its maximal V_C is around 40%, less than others by 10%**.

(d) **Violating the CPC (generally, CPA) constraint does not always being harmful. LR** follows the CPC constraint but loses part of the optimized objectives, especially the

Table 3: The CPC constraint violation ratio of each method.

	06/20/2022-06/26/2022							Avg.	03/13/2023-03/19/2023							Avg.
LR*	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
USCB*	11.6%	11.7%	9.5%	11.9%	11.6%	10.7%	10.3%	11.0%	13.5%	12.6%	12.5%	14.4%	14.9%	11.7%	10.4%	12.9%
PID*	15.1%	20.4%	15.4%	14.9%	17.4%	15.5%	14.3%	16.1%	17.4%	19.5%	23.9%	19.2%	20.3%	16.8%	15.9%	19%
WB*	22.0%	24.0%	21.8%	23.5%	20.7%	21.8%	29.3%	23.3%	26.8%	32.3%	26.3%	26.4%	24.1%	23.9%	25.2%	26.4%
RB	10.4%	12.2%	7.4%	9.7%	10.1%	11.1%	6.9%	**9.7%**	11.3%	12.7%	10.9%	12.4%	11.8%	8.4%	9.6%	**11.0%**

* **LR** [21]. **USCB** [25, 46]. **PID** [49, 56]. **WB** [52].

advertising capability, which contrasts with other methods (as discussed in the previous subsection). It is not easy to say **LR** is better because it uses a rude operation to limit bids and gives up winning some high-valued impressions, which advertisers may not welcome. On the other hand, bid optimization is a multi-constraint problem where feasible solutions are highly dependent on constraints. But setting a reasonable CPC constraint is not straightforward and is coupled with the budget; also, it varies with the market changes. Therefore, in this meaning, the violating campaigns are a signal to note advertisers change their bid optimization setting for better advertising performance.

Table 4: The comparison of violating campaigns*.

	USCB**	**PID****	**WB****	**RB**
The average of V_C	0.1344	0.2583	0.2925	**0.1290**
The maximum of V_C	0.5268	0.8991	0.6723	**0.3939**

* The results are based on the whole experiment.
** **USCB** [25, 46]. **PID** [49, 56]. **WB** [52].

6.4 Ablation experiment

Finally, We test the effort of each algorithm in our method, **RB**. As mentioned in Fig. 3, the tracking part (Algorithm 2) can not run independently because it needs the optimization part (Algorithm 1) to provide references. Consequently, we only conduct the ablation experiment of Algorithm 2 and denote the version of **RB** without Algorithm 2 as **RBWT**. The results are shown in Fig. 6, which are based on auction records from 06/20/2022-06/26/2022.

To sum up, **RBWT** significantly improves the average of $\tilde{G}$ and **CPT**, implying that Algorithm 1 is the main part of our method that affects performance improvement. However, when we move to Std., **RBWT** is not as stable as **RB**, which is not competitive among all methods. Compared **RB** with **RBWT**, although the average performance is not further remarkably improved, the Std. of **RB** is much lower than that of **RBWT** (especially for $\tilde{G}$). It reflects that Algorithm 2 mainly focuses on improving the stability of our method, which is consistent with our original idea in Fig. 3.

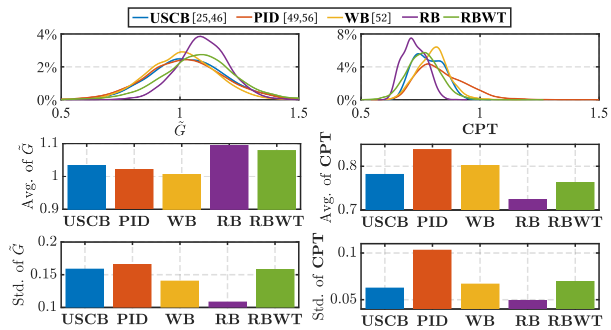

Figure 6: Comparisons of PDF, the average (Avg.), and the standard deviation (Std.) of $\tilde{G}$ and CPT.

7 RELATED WORKS

Optimization-based works: Since RTB emerged in 2009 [32], most researchers regard the bid optimization problem as a constrained optimization problem [18, 33]. Zhang *et al.* [52, 56] proposed a functional optimization framework based on the stochastic process of the auction. Implementing an optimal cost-per-click strategy was also proposed in [24]. Furthermore, a numerical optimal bidding formulation was solved in [49] and used on a real platform. Srinivas and Paul [39], Balseiro *et al.* [6] focused on solving the convergence, regret, and equilibrium problems of the bidding strategy. However, most previous studies have solved the static optimal problem and achieved a static strategy that captures optimality but may lack robustness and adaptability. Although this deficiency can be remedied by introducing modules that predict future trends [42], the resulting solutions exhibit poor real-time ability.

Learning-based works: The root problem in early studies is mainly attributed to market uncertainty and the static strategy, owing to which these previous methods perform considerably below expectations. Recently an increasing number of researchers have attempted to address this issue. Agarwal *et al.*[3] and Xu *et al.*[47] proposed a regularized re-allocate strategy to alter future budget decrements based on the current remnant budget. With the growing popularity of machine learning, Cai *et al.*[11], He *et al.*[25], Jin *et al.*[26], Ren et al.[34], and Wu *et al.*[46] designed various networks and verified their performance on open datasets [30]. Nicolas *et al.*[20] used the recurrent neural network to deal with sequential bidding features. Guan *et al.*[21] used imitation learning to solve the multi-agent cooperative bidding games. Although these methods address a part of the uncertainty, they may not maintain optimality and require large amounts of time and training resources.

Control-based works: Additionally, based on the control theory, previous research has also included a module based on real-time feedback to handle the instability in RTB [52, 54]. This approach first necessitates certain reference trajectories of the interested metrics. It then compares the desired performances and real-time feedback to determine the tracking errors. The last control module adjusts the bidding strategy to follow the references. Zhang *et al.*[54] proposed proportional–integral–derivative (PID) and water-level-based controllers to track cost-per-click. Yang *et al.*[49] also proposed a PID controller to track budget consumption and cost-per-mille. Karlsson [27] analyzed stabilization conditions based on adaptive control and smoothing technology. However, this method cannot function alone, and its performance strongly depends on the designed parameters. Nevertheless, it uses instantaneous information well and can function at high frequencies. By modeling the dynamic process of RTB, the control theory can be used to analyze and grasp the robustness and adaptability of bid optimization.

8 CONCLUSION

We propose a bid optimization methodology for e-commerce platforms in real-time multi-constraint online advertising. It can be implemented online as it employs a rollout mechanism and a sampling strategy on the logs of historical bid records. To account for the uncertainty stemming from the market, we design a tracking algorithm to stabilize the bidding process during two adjoint intervals and achieve optimal tracking performance. Numerical experiments on real datasets verify its performance. In the future, introducing predictions to achieve more foresight in the face of market fluctuations will be of great interest.

A EMPIRICAL SAMPLING SIZE EXPERIMENTS

We conduct experiments to help us set the sampling size N in Algorithm 1, balancing the consumption and accuracy.

A.1 Experiment settings

A.1.1 Experiment scenario. Following the business service provided on Taobao, the objective is maximizing the expected **Gross merchandise volume (GMV)**. Constraints are **budget** B and **cost-per-click (CPC)** C. Thus, for the original bid optimization problem (1), $v(\mathbf{x}) = \hat{p}^{\mathrm{CTR}}(\mathbf{x})\hat{p}^{\mathrm{CVR}}(\mathbf{x})i_p$, $k = 1$, $v_a(\mathbf{x}) = \hat{p}^{\mathrm{CTR}}(\mathbf{x})$, where i_p is the item price. Based on Theorem 4.3, the bidding strategy is $b(\mathbf{x}_t) = (\hat{p}^{\mathrm{CVR}}(\mathbf{x}_t)i_p + \beta C)/(\alpha + \beta)$.

A.1.2 Dataset. Taobao provides an anonymous dataset. We randomly selected 1000 campaigns in the same day, totaling over 40 million records. Each record consists of

$$\{year/month/day/hour/minute/second, \hat{p}^{\mathrm{CTR}}, \hat{p}^{\mathrm{CVR}}, i_p, \hat{w}\},$$

where the string 'year/month/day/hour/minute/second' is the impression occurrence time. For an auctioned impression, $\hat{w}$ is the winning price, i_p is the item price, $\hat{p}^{\mathrm{CTR}}$ and $\hat{p}^{\mathrm{CVR}}$ are the estimated click-through rate and conversion rate, respectively.

A.1.3 Experiment pipeline. The experiment is based on *bootstrap*[15]. For each ad campaign, the optimal dual solution of (6) solved by all its records is regarded as the truth, denoted by $\{\alpha^*, \beta^*\}$. Then, to test the impact of sampling sizes N, six groups are set,

$$N = \{1000, 5000, 10000, 20000, 30000, 50000\}.$$

For each N, we uniformly sample records and solve (6), denoted as $\{\alpha_N, \beta_N\}$. Besides, each ad campaign will be sampled and solved 1000 times for each sampling size N.

A.1.4 Metrics. Since α and β together determine the bid price in Theorem 4.3, it is not straightforward to compare them separately. Alternatively, we compare bids from $\{\alpha^*, \beta^*\}$ and $\{\alpha_N, \beta_N\}$ and define the normalized bid price error as

$$E_b = (b_N - b^*)/b^*,$$

where b^* is the bid price comes from the truth $\{\alpha^*, \beta^*\}$, b_N comes from $\{\alpha_N, \beta_N\}$. We compare the following metrics of different N

(a) The average (Avg.) and standard deviation (Std.) of E_b.
(b) Solving resource consumption, consisting of

$$\bar{T}_N = T_N/T_{1000},\ \bar{G}^{\mathrm{C}}_N = G^{\mathrm{C}}_N/G^{\mathrm{C}}_{1000},\ \bar{G}^{\mathrm{M}}_N = G^{\mathrm{M}}_N/G^{\mathrm{M}}_{1000},$$

where $\bar{T}_N$, $\bar{G}^{\mathrm{C}}_N$ and $\bar{G}^{\mathrm{M}}_N$ are the average of normalized solving time, CPU consumption and memory consumption with different N, respectively. Denominators T_{1000}, G^{C}_{1000} and G^{M}_{1000} are the averages of each campaign with $N = 1000$.

A.2 Results discussion

All experiments are implemented on Dataworks [14], and the linear programming solver is MindOpt [1]. In summary:

(a) In Table 5, when $N = 1000$, Avg.=0.1592 means the average of bid prices is higher than its truth by 16%. As N increases, Avg. of E_b decreases as expected. Avg. is less than 0.05 as $N \geq 10000$; empirically, it is enough for usage. Besides, Std. decreases as N increases, implying it tends to be stable.
(b) Table 6 compares solving resource consumption. Undisputedly, increasing N would exponentially increase the solving resource consumption. Specifically, it takes around several seconds, 0.2 Core∗Minute CPU, 0.3 GB ∗Minute memory to solve the problem when $N = 1000$; however, it takes nearly half an hour, more than 25 Core∗Minute CPU and over 40 GB∗Minute memory, to solve it with $N = 50000$.

Thus, we choose $N = 10000$ as an empirically feasible choice for Algorithm 1 in this paper. Meanwhile, we make some custom designs for the solver used online, such as pre-allocated space, etc.; the solving time of an ad campaign is about 10ms when $N = 10000$. In addition, for platforms like Taobao, we count that in the case of $N = 10000$, the sampled records only account for about 3% of the original, dramatically saving storage space.

Table 5: The average and the standard deviation of E_b.

N	1000	5000	10000	20000	30000	50000
Avg. of E_b	0.1592	0.0926	0.0487	0.0359	0.0262	0.0183
Std. of E_b	0.1551	0.0724	0.0212	0.0133	0.0099	0.0035

Table 6: Comparison of solving resource consumption.

N	1000	5000	10000	20000	30000	50000
$\bar{T}_N$	1.0000	1.7310	4.6559	13.4376	32.5179	237.1962
$\bar{G}^{\mathrm{C}}_N$	1.0000	1.3522	3.1517	8.8527	19.1295	125.3263
$\bar{G}^{\mathrm{M}}_N$	1.0000	1.3793	3.6552	10.1379	144.9310	430.5273

REFERENCES

[1] Alibaba Damo Academy. 2022. MindOpt Optimization Suite. Retrieved 2022 from https://solver.damo.alibaba.com/htmlpages/page#/en

[2] Deepak Agarwal, Datong Chen, Long-ji Lin, Jayavel Shanmugasundaram, and Erik Vee. 2010. Forecasting high-dimensional data. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. 1003–1012.

[3] D. Agarwal, S. Ghosh, W. Kai, and S. You. 2014. Budget pacing for targeted online advertisements at LinkedIn. In *Knowledge Discovery and Data Mining*.

[4] Anonymous. 2023. Supplement. Retrieved 2023 from https://anonymous.4open.science/r/supplement_SIGMOD-F4B0/supplement.pdf

[5] Santiago Andrés Azcoitia and Nikolaos Laoutaris. 2022. A survey of data marketplaces and their business models. *ACM SIGMOD Record* 51, 3 (2022), 18–29.

[6] Santiago R Balseiro and Yonatan Gur. 2019. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science* 65, 9 (2019), 3952–3968.

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

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

[9] The Interactive Advertising Bureau. 2023. *Internet Advertising Revenue Report: Full Year 2022.* Retrieved Apr. 12, 2023 from https://www.iab.com/insights/internet-advertising-revenue-report-full-year-2022/

[10] Brian S Cade and Barry R Noon. 2003. A gentle introduction to quantile regression for ecologists. *Frontiers in Ecology and the Environment* 1, 8 (2003), 412–420.

[11] 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.

[12] Badrish Chandramouli, Jonathan Goldstein, and Songyun Duan. 2012. Temporal analytics on big data for web advertising. In *2012 IEEE 28th international conference on data engineering*. IEEE, 90–101.

[13] Hana Choi, Carl F Mela, Santiago R Balseiro, and Adam Leary. 2020. Online display advertising markets: A literature review and future directions. *Information Systems Research* 31, 2 (2020), 556–575.

[14] Alibaba Cloud. 2022. DataWorks. Retrieved 2022 from https://www.alibabacloud.com/help/en/dataworks/latest/what-is-dataworks

[15] Anthony Christopher Davison and David Victor Hinkley. 1997. *Bootstrap methods and their application.* Number 1. Cambridge university press.

[16] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. 2013. *Feedback control theory.* Courier Corporation.

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

[18] 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*. 251–260.

[19] Google.com. 2022. Google Ads Help. Retrieved 2022 from https://support.google.com/google-ads

[20] 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.

[21] 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.

[22] 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).

[23] Kai Han, Benwei Wu, Jing Tang, Shuang Cui, Cigdem Aslay, and Laks VS Lakshmanan. 2021. Efficient and effective algorithms for revenue maximization in social advertising. In *Proceedings of the 2021 international conference on management of data*. 671–684.

[24] 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*.

[25] 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.

[26] 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.

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

[28] Javen Kennedy, Pranav Subramaniam, Sainyam Galhotra, and Raul Castro Fernandez. 2022. Revisiting Online Data Markets in 2022: A Seller and Buyer Perspective. *ACM SIGMOD Record* 51, 3 (2022), 30–37.

[29] Barrie Kersbergen, Olivier Sprangers, and Sebastian Schelter. 2022. Serenade-low-latency session-based recommendation in e-commerce at scale. In *Proceedings of the 2022 International Conference on Management of Data*. 150–159.

[30] Hairen Liao, Lingxiao Peng, Zhenchuan Liu, and Xuehua Shen. 2014. iPinYou global rtb bidding algorithm competition dataset. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. 1–6.

[31] 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.

[32] Shanmugavelayutham Muthukrishnan. 2009. Ad exchanges: Research issues. In *International workshop on internet and network economics*. Springer, 1–12.

[33] Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster Provost. 2012. Bid optimizing and inventory scoring in targeted online advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 804–812.

[34] 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.

[35] Philippe Roy, René Laprise, and Philippe Gachon. 2016. Sampling errors of quantile estimations from finite samples of data. *arXiv preprint arXiv:1610.03458* (2016).

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

[37] Jing Tang, Keke Huang, Xiaokui Xiao, Laks VS Lakshmanan, Xueyan Tang, Aixin Sun, and Andrew Lim. 2019. Efficient approximation algorithms for adaptive seed minimization. In *Proceedings of the 2019 International Conference on Management of Data*. 1096–1113.

[38] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *Proceedings of the 2018 International Conference on Management of Data*. 991–1005.

[39] 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.

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

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

[42] Jun Wang, Weinan Zhang, Shuai Yuan, et al. 2017. Display advertising with real-time bidding (RTB) and behavioural targeting. *Foundations and Trends® in Information Retrieval* 11, 4-5 (2017), 297–435.

[43] 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.

[44] 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.

[45] 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.

[46] 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*. 1443–1451.

[47] J. Xu, K. C. Lee, W. Li, H. Qi, and Q. Lu. 2015. Smart Pacing for Effective Online Ad Campaign Optimization. In *ACM*.

[48] Zhiqiang Xu, Dong Li, Weijie Zhao, Xing Shen, Tianbo Huang, Xiaoyun Li, and Ping Li. 2021. Agile and accurate CTR prediction model training for massive-scale online advertising systems. In *Proceedings of the 2021 international conference on management of data*. 2404–2409.

[49] 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.

[50] 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.

[51] 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.

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

[53] 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).

[54] Weinan Zhang, Yifei Rong, Jun Wang, Tianchi Zhu, and Xiaofan Wang. 2016. Feedback control of real-time display advertising. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 407–416.
[55] 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.
[56] 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.
[57] Yipeng Zhang, Yuchen Li, Zhifeng Bao, Baihua Zheng, and HV Jagadish. 2021. Minimizing the regret of an influence provider. In *Proceedings of the 2021 International Conference on Management of Data*. 2115–2127.
[58] 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.