A Pratical Guide to Budget Pacing Algorithms in Digital Advertising

For Engineers

Yuanlong Chen

PREFACE

A typical real-time ad-serving funnel comprises ad targeting, conversion (e.g. click through rate) modeling, budget pacing (bidding), and auction processes. While there is a wealth of research and articles on ad targeting and conversion modeling, budget pacing—a crucial component—remains underexplored in existing literature. This book aims to provide engineers with a practical yet comprehensive introduction to budget pacing algorithms within the digital advertising domain. The book is structured as follows:

In Part I, we introduce foundational concepts in the digital advertising business, along with preliminary knowledge essential for understanding the subsequent chapters. We begin with a brief introduction to the history of digital advertising. Next, we cover some basics of programmatic ads, including the concepts of CPM, CPC, and CPA ads. The entire ad-serving funnel is then briefly discussed to illustrate how ads are served in real time. The pipeline presented focuses on first-party ads (e.g., ads on YouTube or Instagram), though the serving pipeline for DSPs is similar. Additionally, we address basic optimization techniques, auction mechanism design, and other related preliminary topics that will be referenced throughout the book. Readers already familiar with these subjects may choose to skip this section and proceed directly to Part II.

In Part II, we discuss various pacing methods under standard second price auction. Two main bidding products, max delivery and cost cap, are used as examples to demonstrate the concepts of these pacing methods. Nevertheless, the underlying principles introduced here are applicable to other problems as well. We first provide a rigorous mathematical formulation of both the max delivery and cost cap problems. In the subsequent sections, we discuss various pacing algorithms commonly adopted in the industry, including throttling, PID controllers, MPC controllers, online adaptive optimal control, and deep reinforcement learning. For each approach, we explain the motivation, introduce the basic background, and describe how it can be applied to bidding problems such as max delivery and cost cap. Additionally, we discuss the pros and cons of each approach, enabling readers to select the most suitable method for real-world applications based on their specific business needs. For some algorithms, pseudo-code and simple implementations are also provided to give readers a practical understanding of how to implement them in their daily work.

In Part III, we demonstrate how the pacing frameworks introduced in Part II can be applied to various other business scenarios. Topics include the initialization of campaign bids, bidding under different auction mechanisms (e.g., first-price auctions, where bid shading is required), bid optimization for multi-constraint problems (e.g., campaigns delivered across different placements or channels such as first-party and third-party platforms, or campaign groups where multiple campaigns share the same budget), deep funnel conversion problems (e.g., bid optimization for post-conversion events such as retention), common brand advertisements with reach and frequency requirements, and the over-delivery problem.

Budget optimization in digital advertising is a broad and complex topic. This little book primarily aims to provide engineers in the field with a comprehensive overview of the landscape of budget pacing algorithms. It does not attempt to cover every detail of budget pacing. For better readability, we omit some theoretical aspects, such as regret analysis and equilibrium analysis. Readers interested in these topics are encouraged to refer to the academic papers mentioned throughout the book for more in-depth information.

Y. Chen

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Part I

Some Basics

CHAPTER 1

A BRIEF HISTORY OF DIGITAL ADS

Brief History Intro

Introduction

Digital advertising has undergone a remarkable transformation since its inception, evolving from simple banner ads to sophisticated programmatic systems powered by real-time data. At the heart of this evolution is Real-Time Bidding (RTB), an innovation that has revolutionized the way advertisers and publishers interact. RTB operates alongside key players such as Demand-Side Platforms (DSPs), Supply-Side Platforms (SSPs), and major in-house bidding systems like Google Ads and Facebook Ads Manager. This chapter explores the history of digital advertising, with a particular focus on the rise of RTB and its integration into a broader programmatic ecosystem.

The Early Days of Digital Advertising

The first era of digital advertising began in the mid-1990s with the advent of the internet. Banner ads, such as the iconic AT&T ad on HotWired in 1994, marked the start of online monetization. During this period, advertisers purchased ad space directly from publishers, with little data available to inform decisions. As internet adoption grew, ad networks emerged to connect advertisers with publishers more efficiently. These networks aggregated inventory but lacked sophisticated targeting capabilities, leading to inefficiencies and limited personalization.

The Rise of Programmatic Advertising

The introduction of programmatic advertising in the early 2000s addressed many of the shortcomings of traditional models. Automated systems began replacing manual negotiations, enabling advertisers to target audiences based on demographic, geographic, and behavioral data. This innovation paved the way for the creation of DSPs and SSPs.

Demand-Side Platforms (DSPs)

DSPs provide advertisers with a centralized platform to manage and optimize ad campaigns across multiple channels. By leveraging advanced algorithms and real-time data, DSPs empower advertisers to bid on impressions that align with their target audience and campaign objectives.

Supply-Side Platforms (SSPs)

On the publisher side, SSPs enable efficient management of ad inventory. SSPs connect publishers to multiple ad exchanges and DSPs, ensuring maximum revenue through competitive bidding. Together, DSPs and SSPs form the backbone of the programmatic advertising ecosystem.

The Advent of Real-Time Bidding (RTB)

RTB emerged in 2009 as a game-changer in programmatic advertising. Unlike earlier methods that involved bulk purchasing of ad space, RTB allows advertisers to bid on individual impressions in real time. Key milestones in RTB history include:

- **2009:** Launch of DoubleClick Ad Exchange by Google, introducing the first large-scale RTB platform.
- **2011:** Facebook introduced its ad exchange (FBX), extending RTB capabilities to social media advertising.
- **2013:** Mobile RTB gained prominence, reflecting the rapid growth of mobile internet usage.
- **2015:** Header bidding strategies allowed publishers to maximize revenue by offering inventory to multiple exchanges simultaneously.

In-House Real-Time Bidding Systems

Major platforms like Google and Facebook have developed their own proprietary bidding systems to capitalize on their vast user bases and data resources:

Google Ads

Google Ads integrates RTB capabilities into its ecosystem, allowing advertisers to bid on search, display, and video ads. Its advanced targeting and optimization features make it a dominant force in digital advertising.

Facebook Ads Manager

Facebook introduced FBX to integrate RTB into its platform. Although FBX was later retired, Facebook Ads Manager continues to provide advertisers with precise audience targeting and real-time campaign optimization.

Challenges and Innovations in Digital Advertising

While digital advertising has achieved remarkable success, it faces ongoing challenges such as data privacy concerns, ad fraud, and transparency issues. Regulations like GDPR and CCPA have reshaped the industry, emphasizing the need for responsible data usage. Simultaneously, emerging technologies such as artificial intelligence and blockchain are driving innovation, addressing these challenges, and enhancing efficiency.

Conclusion

The history of digital advertising reflects a journey of continuous innovation, from static banners to dynamic, data-driven systems like RTB. Understanding this evolution highlights the transformative impact of programmatic technologies and the importance of platforms like DSPs, SSPs, and in-house solutions from tech giants like Google and Facebook. As digital advertising continues to evolve, RTB remains a cornerstone, shaping the future of how brands connect with audiences.

CHAPTER 2

ADVERTISING BASICS

Basic concepts in digital advertising

CHAPTER 3

BIDDING PRODUCTS

Introduction to some popular bidding products

CHAPTER 4

OPTIMIZATION TECHNIQUES

Some basics of optimization techniques

CHAPTER 5

AUCTION MECHANISMS

Introduction to some popular auction mechanisms

CHAPTER 6

EXPERIMENT FRAMEWORK

Introduction to A/B testing framework in advertisering business

Part II

Pacing Algorithms

CHAPTER 1

BIDDING PROBLEM FORMULATION

In this chapter, we provide a rigorous mathematical formulation of two primary bidding problems, namely max delivery and cost cap, in the context of repeated auction settings. We then employ the primal-dual method to derive the optimal bidding formulas. These results serve as the foundation for designing online control algorithms, which will be explored in the subsequent chapters.

1 Max Delivery

In the Max Delivery setting, advertisers set up a campaign with a specified budget B. The objective is to optimize the performance of the ad campaign while adhering to this budget constraint. Assuming this is a CPC (Cost-Per-Click) daily campaign and operates under the standard Second Price Auction framework, the Max Delivery problem can be formulated as the following optimization problem:

$$\begin{aligned} \max_{x_t \in \{0,1\}} \quad & \sum_{t=1}^{T} x_t \cdot r_t \\ \text{s.t.} \quad & \sum_{t=1}^{T} x_t \cdot c_t \leq B \end{aligned} \tag{1.1}$$

Here, T represents the total number of auction opportunities within a day. For the t-th auction:

- r_t is the predicted click-through rate (CTR).
- c_t denotes the cost (in a second-price auction, this corresponds to the highest effective CPM [eCPM]).
- x_t is a decision variable indicating whether we win the auction.

Under the rules of a second-price auction, $x_t = 1$ if and only if our bid per impression exceeds the highest competing bid, mathematically expressed as:

$$x_t = \mathbb{1}_{\{b_t > c_t\}}.$$

We assume that both the sequences $\{r_t\}$ and $\{c_t\}$ follows some unknown independent and identically distributed (*i.i.d.*) distribution, such as a log-normal distribution.

Optimal Solution to Max Delivery Problem

It is challenging to solve this problem directly. Instead of addressing it in the primal space, we apply the primal-dual method to transform it into the dual space. The Lagrangian of (1.1) is given by:

$$\mathcal{L}(x, \lambda) = \sum_{t=1}^{T} x_t \cdot r_t - \lambda \cdot \left(\sum_{t=1}^{T} x_t \cdot c_t - B \right)$$

The dual is expressed as:

$$\min_{\lambda \geq 0} \mathcal{L}^*(\lambda) = \min_{\lambda \geq 0} \max_{x_t \in \{0,1\}} \mathcal{L}(x, \lambda).$$

We can rewrite $\mathcal{L}(x, \lambda)$ as:

$$\mathcal{L}(x, \lambda) = \sum_{t=1}^{T} x_t \cdot (r_t - \lambda c_t) + \lambda B.$$

To maximize $\mathcal{L}(x,\lambda)$, we set $x_t = 1$ whenever $r_t - \lambda c_t > 0$, and $x_t = 0$ otherwise. Consequently, $\mathcal{L}^*(\lambda) = \max_{x_t \in \{0,1\}} \mathcal{L}(x,\lambda)$ becomes:

$$\mathcal{L}^*(\lambda) = \sum_{t=1}^{T} (r_t - \lambda c_t)_+ + \lambda B,$$

where $(z)_+ = \mathbb{1}_{\{z>0\}} \cdot z$ is the ReLU function. Therefore, the dual problem is:

$$\min_{\lambda \geq 0} \mathcal{L}^*(\lambda) = \min_{\lambda \geq 0} \sum_{t=1}^{T} \left[(r_t - \lambda c_t)_+ + \lambda \cdot \frac{B}{T} \right]. \tag{1.2}$$

Suppose the problem is feasible and

$$\lambda^* = \arg\min_{\lambda \geq 0} \mathcal{L}^*(\lambda)$$

The KKT conditions indicate that λ^* is the optimal dual variable that satisfies the budget constraint:

$$\sum_{t=1}^{T} x_t \cdot c_t = B$$

The optimal bid per impression is determined as:

$$b_t^* = \frac{r_t}{\lambda^*}$$

The optimal bid per click is given by

$$b_{click}^* = \frac{1}{\lambda^*} \tag{1.3}$$

Quick summary of our main results

The optimal bid per click for (1.1) in the stochastic setting is a constant bid:

$$b_{click}^* = \frac{1}{\lambda^*}$$

Suppose supply is sufficient (T big enough), the constant optimal bid b_{click}^* is the bid per click that exactly depletes the budget, it also suggests that the amount of budget depleted within a time interval is proportional to the number of auction opportunities, i.e.,

$$\sum_{\tau \leq t \leq \tau + d\tau} x_t c_t \approx \# \text{ of auction opportunities in } (\tau, \tau + d\tau).$$

2 Cost Cap

Cost Cap is a product designed for price-sensitive advertisers. In addition to specifying a budget B, the advertiser defines a cost cap C, which sets an upper limit on the average cost per result. This ensures that the average cost per result does not exceed the specified cap. Using the notation from the previous section, the cost cap problem for a CPC daily campaign can be formulated as follows:

$$\begin{aligned} \max_{x_t \in \{0,1\}} \quad & \sum_{t=1}^{T} x_t \cdot r_t \\ \text{s.t.} \quad & \sum_{t=1}^{T} x_t \cdot c_t \leq B \\ & \frac{\sum_{t=1}^{T} x_t \cdot c_t}{\sum_{t=1}^{T} x_t \cdot r_t} \leq C \end{aligned} \tag{1.4}$$

We make the same assumption that the sequences $\{r_t\}$ and $\{c_t\}$ follows some unknown *i.i.d.* distribution.

Optimal Solution to Cost Cap Problem

We apply the primal-dual method to solve (1.4), as was done for the maximum delivery problem. The key difference is that we now have two constraints. The Lagrangian for this problem is given by:

$$\mathcal{L}(x, \lambda, \mu) = \sum_{t=1}^{T} x_t \cdot r_t - \lambda \cdot \left(\sum_{t=1}^{T} x_t \cdot c_t - B \right) - \mu \cdot \left[\sum_{t=1}^{T} x_t \cdot c_t - C \cdot \left(\sum_{t=1}^{T} x_t \cdot r_t \right) \right]$$

The dual is expressed as:

$$\min_{\lambda \geq 0, \mu \geq 0} \mathcal{L}^*(\lambda, \mu) = \min_{\lambda \geq 0, \mu \geq 0} \max_{x_t \in \{0,1\}} \mathcal{L}(x, \lambda, \mu).$$

Note that $\mathcal{L}(x, \lambda, \mu)$ can be rewritten as:

$$\mathcal{L}(x, \lambda, \mu) = \sum_{t=1}^{T} x_t \cdot (r_t - \lambda c_t - \mu c_t + \mu C r_t) + \lambda B.$$

Similarly, to maximize $\mathcal{L}(x, \lambda, \mu)$, we set $x_t = 1$ whenever $r_t - \lambda c_t - \mu c_t + \mu C r_t > 0$, and $x_t = 0$ otherwise. $\mathcal{L}^*(\lambda, \mu) = \max_{x_t \in \{0,1\}} \mathcal{L}(x, \lambda, \mu)$ then becomes:

$$\mathcal{L}^*(\lambda, \mu) = \sum_{t=1}^{T} (r_t - \lambda c_t - \mu c_t + \mu C r_t)_+ + \lambda B,$$

where $(\cdot)_+$ again is the ReLU function. The dual problem of (1.4) is:

$$\min_{\lambda \geq 0, \mu \geq 0} \mathcal{L}^*(\lambda, \mu) = \min_{\lambda \geq 0, \mu \geq 0} \sum_{t=1}^{T} \left[(r_t - \lambda c_t - \mu c_t + \mu C r_t)_+ + \lambda \cdot \frac{B}{T} \right]. \tag{1.5}$$

Suppose we have feasible solution to this problem

$$\lambda^*, \mu^* = \underset{\lambda \geq 0, \mu \geq 0}{\arg\min}\, \mathcal{L}^*(\lambda, \mu)$$

The optimal bid per impression is determined as:

$$b_t^* = \frac{1 + \mu^* C}{\lambda^* + \mu^*} \cdot r_t$$

The optimal bid per click is given by

$$b_{click}^* = \frac{1}{\lambda^* + \mu^*} + \frac{\mu^*}{\lambda^* + \mu^*} \cdot C = \frac{\lambda^*}{\lambda^* + \mu^*} \cdot \frac{1}{\lambda^*} + \frac{\mu^*}{\lambda^* + \mu^*} \cdot C \tag{1.6}$$

Setting $\alpha = \lambda^*/(\lambda^* + \mu^*)$, we have

$$b_{click}^* = \alpha \cdot \frac{1}{\lambda^*} + (1 - \alpha) \cdot C \tag{1.7}$$

Note that $1/\lambda^*$ is the optimal bid in max delivery without considering the cost constraint. Therefore, the optimal bid for the cost cap is simply a linear combination of the unconstrained max delivery bid and the cost cap bid.

Quick summary of our main results for cost cap problem

The optimal bid per click for cost cap problem (1.4) in the stochastic setting is a constant, more specifically, simply a linear combination of the unconstrained max delivery bid and the cost cap bid:

$$b_{click}^* = \alpha \cdot \frac{1}{\lambda^*} + (1 - \alpha) \cdot C$$

where $\alpha = \frac{\lambda^*}{\lambda^* + \mu^*}$.

CHAPTER 2

THROTTLE-BASED PACING

In this chapter, we dicuss budget pacing algorithms via throttling. In the context of budget pacing, throttling refers to a mechanism that controls a campaign's participation in real-time auctions based on its actual budget spending relative to a target budget, which is determined by the supply pattern. This approach ensures that the ad campaign's budget is distributed in alignment with the supply pattern over its duration, thereby optimizing the campaign's performance.

1 Probabilistic Throttling

In throttle-based pacing, ad campaigns participate in online auctions with a fixed, pre-defined bid. We first consider a daily max-delivery campaign. If the fixed bid is set inappropriately and it's relatively high, it is highly likely that the campaign will win the majority of auction opportunities early on. As a result, the campaign may overspend at the start, rapidly exhausting the budget well before the end of the day. The probabilistic throttling algorithm is a mechanism used to control the participation of a campaign in real-time auctions based on its current spending relative to a target budget. The algorithm operates by dynamically adjusting a participation probability $p(t)$ that determines whether the campaign enters or skips a given auction. If the campaign is currently over-delivered (i.e., actual spending exceeds the expected spending), $p(t)$ is decreased, making it less likely to participate in the current auction, and vice versa when the campaign is under-delivered. This probabilistic control ensures that the campaign's budget is reasonably distributed over its time horizon while maximizing performance opportunities.

Modifications can be made to adapt this algorithm for a cost cap setting, where an additional performance constraint is imposed to ensure that the campaign achieves its goals within a specified cost threshold.

Throttling for Max-Delivery Campaign

From the discussion in Equation 1, still assumming i.i.d. distributions of the conversion rates $\{r_t\}$ and costs $\{c_t\}$, we know that the optimal budget allocation is achieved when the budget for each duration is distributed proportionally to the total number of eligible auction opportunities (supply) available during that period.

Suppose the prediction model estimates there are T auction opportunities for this campaign within a day(in practice, T is typically derived by analyzing historical time series data, and the prediction accuracy of T at the campaign level may vary, which can degrade the performance of the pacing algorithm. Some online adjustments might be implemented to reduce the prediction noise; however, we will not discuss those techniques here. For simplicity, we assume the prediction is perfect). At the t-th auction, with a perfect pacing algorithm, the spend should be $\alpha(t) = \frac{t}{T} \cdot B$, where B is the total budget. If the actual spend $S(t) > \alpha(t)$, meaning pacing is ahead of schedule, we should slow down the pacing rate. An intuitive approach is to set a participation probability $p(t)$, which determines the likelihood of the campaign participating in the t-th auction. In the case of over-delivery, we lower $p(t)$ to reduce the chance of participating in the auction, thereby decreasing the likelihood of spending during this round. Mathematically, we can update $p(t)$ by multiplying it by $1 - \lambda_t$, where $\lambda_t > 0$ is a control parameter to adjust the throttling level. Conversely, if the campaign is under-delivered, we increase $p(t)$ by multiplying it by $1 + \lambda_t$. Mathematically, the update rule of $p(t)$ can be expressed as follows:

$$p(t) = \begin{cases} \min\{p(t-1) \cdot (1 + \lambda_t), 1\} & \text{if } S(t) \leq \alpha(t), \\ \max\{p(t-1) \cdot (1 - \lambda_t), 0\} & \text{if } S(t) > \alpha(t). \end{cases}$$

This motivates the following Algorithm 1:

Algorithm 1 Throttling-based Budget Pacing Algorithm

Require: B: Total budget of the campaign
Require: T: Total number of auction opportunities
Require: t: Current auction round
Require: $S(t)$: Spend so far at t-th auction
Require: $p(t)$: Throttling probability at t-th auction
Require: $\{\lambda_t\}$: Control parameters for throttling adjustment

Initialize $p(0) \leftarrow 1.0$ and $S(0) \leftarrow 0.0$
for each auction at t-th auction **do**
    Calculate target spend: $target\_spend\alpha(t) \leftarrow \frac{t}{T} \times B$
    if $S(t) \leq \alpha(t)$ **then**
        Increase throttling probability: $p(t) \leftarrow \min\{1.0, p(t) \cdot (1 + \lambda_t\}$
    else
        Decrease throttling probability: $p(t) \leftarrow \max\{0.0, p(t) \cdot (1 - \lambda_t\}$
    end if
    Generate a random number $r \in [0, 1]$
    if $r \leq p(t)$ **then**
        Participate in the auction and get the spend in current auction c_t
        Update spend: $S(t) \leftarrow S(t-1) + c_t$
    else
        Skip the auction
    end if
end for

In practice, to simplify the implementation, we may set λ_t as a constant, e.g. 10%, as in [1]. Also, there is no need to update $p(t)$ for every auction, we may set the update granularity to, say, 1 minute. More technical implementation details could be found in [1]. The regret analysis and the optimality of throttle-based pacing can be found in [3], which also includes a comparison between throttle-based pacing and bid-based pacing, both of which we will introduce in the subsequent sections.

Throttling for Cost Cap Campaign

More technical details on cost cap throttling [6]

CHAPTER 3

PID CONTROLLER

PID controller

1 Introduction to PID Controllers

A Proportional-Integral-Derivative (PID) controller is a widely used feedback control mechanism in industrial and engineering systems. It is designed to maintain a desired output by minimizing the error $e(t)$, which is the difference between a desired setpoint $r(t)$ and the measured process variable $y(t)$. The PID controller achieves this by adjusting the control input $u(t)$ based on three terms: proportional, integral, and derivative. The primary motivation for using a PID controller is its ability to regulate dynamic systems efficiently by balancing fast response, minimal steady-state error, and robustness to disturbances. PID controllers are versatile and can be tuned to meet specific performance requirements in diverse applications, such as temperature control, motor speed regulation, and process automation.

Mathematical Details

The output of a PID controller, $u(t)$, is given by:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt},$$

where:

- $e(t) = r(t) - y(t)$ is the error signal,
- K_p is the proportional gain, controlling the response proportional to the error,
- K_i is the integral gain, reducing steady-state error by integrating the error over time,
- K_d is the derivative gain, predicting future error by calculating the rate of change of the error.

Components of a PID Controller

1. **Proportional Term**: $K_p e(t)$ provides an immediate response proportional to the current error. However, it may not fully eliminate the steady-state error.
2. **Integral Term**: $K_i \int_0^t e(\tau)\, d\tau$ accumulates past errors, addressing steady-state error by applying corrective action based on the error history.
3. **Derivative Term**: $K_d \frac{de(t)}{dt}$ predicts future errors by responding to the rate of change of the error, improving stability and damping oscillations.

PID controllers are simple to implement, robust, and effective for a wide range of systems. With proper tuning of K_p, K_i, and K_d, they can balance speed, stability, and accuracy in dynamic environments.

2 PID Contoller in Max Delivery

PID controller approach: [7]

3 PID Contoller in Cost Cap

CHAPTER 4

MPC CONTROLLER

MPC controller

CHAPTER 5

DUAL ONLINE GRADIENT DESCENT

Online Gradient Descent in dual space

1 Max Delivery

ref: [4] [2]

CHAPTER 6

DEEP REINFORCEMENT LEARNING

Deep Reinforcement Learning in auto-bidding

Part III

Miscellaneous

CHAPTER 1

INITIALIZATION OF CAMPAIGN BID

Initial Bid

CHAPTER 2

BID SHADING

Bid Shading in first price auction

1 Bid Shading via Surplus Maximization

ref: [5]

CHAPTER 3

MULTI-CHANNEL DELIVERY

Multi-Channel Delivery problem.

CHAPTER 4

CAMPAIGN GROUP OPTIMIZATION

Campaign Group Optimization

CHAPTER 5

DEEP CONVERSION

Deep conversion(retention) optimization.

CHAPTER 6

REACH AND FREQUENCY

Brand ads, reach and frequency

CHAPTER 7

OVER-DELIVERY PROBLEM

Over Delivery

BIBLIOGRAPHY

[1] Deepak Agarwal, Souvik Ghosh, Kai Wei, and Siyu You. "Budget pacing for targeted online advertisements at LinkedIn". In: KDD '14 (2014), pp. 1613–1619.

[2] Santiago R Balseiro and Yonatan Gur. "Learning in repeated auctions with budgets: Regret minimization and equilibrium". In: *Management Science* 65.9 (2019), pp. 3952–3968.

[3] Zhaohua Chen, Chang Wang, Qian Wang, Yuqi Pan, Zhuming Shi, Zheng Cai, Yukun Ren, Zhihua Zhu, and Xiaotie Deng. "Dynamic budget throttling in repeated second-price auctions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 9. 2024, pp. 9598–9606.

[4] Yuan Gao, Kaiyu Yang, Yuanlong Chen, Min Liu, and Noureddine El Karoui. "Bidding agent design in the linkedin ad marketplace". In: *arXiv preprint arXiv:2202.12472* (2022).

[5] Shengjun Pan, Brendan Kitts, Tian Zhou, Hao He, Bharatbhushan Shetty, Aaron Flores, Djordje Gligorijevic, Junwei Pan, Tingyu Mao, San Gultekin, et al. "Bid shading by win-rate estimation and surplus maximization". In: *arXiv preprint arXiv:2009.09259* (2020).

[6] Jian Xu, Kuang-chih Lee, Wentong Li, Hang Qi, and Quan Lu. "Smart pacing for effective online ad campaign optimization". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 2217–2226.

[7] Weinan Zhang, Yifei Rong, Jun Wang, Tianchi Zhu, and Xiaofan Wang. "Feedback control of real-time display advertising". In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. 2016, pp. 407–416.