

Bidding Agent Design in the LinkedIn Ad Marketplace

Yuan Gao
yugao@linkedin.com
LinkedIn Corporation
Mountain View, CA, USA

Kaiyu Yang
kyang@linkedin.com
LinkedIn Corporation
Mountain View, CA, USA

Yuanlong Chen
yuachen@linkedin.com
LinkedIn Corporation
Mountain View, CA, USA

Min Liu
mliu@linkedin.com
LinkedIn Corporation
Mountain View, CA, USA

Noureddine El Karoui
nkarouiprof@gmail.com
Work done in part while at LinkedIn,
Inc. and UC, Berkeley
Berkeley, CA, USA

ABSTRACT

We establish a general optimization framework for the design of automated bidding agent in dynamic online marketplaces. It optimizes solely for the buyer's interest and is agnostic to the auction mechanism imposed by the seller. As a result, the framework allows, for instance, the joint optimization of a group of ads across multiple platforms each running its own auction format. Bidding strategy derived from this framework automatically guarantees the optimality of budget allocation across ad units and platforms. Common constraints such as budget delivery schedule, return on investments and guaranteed results, directly translates to additional parameters in the bidding formula. We share practical learnings of the deployed bidding system in the LinkedIn ad marketplace based on this framework.

CCS CONCEPTS

• **Information systems** → **Online advertising; Sponsored search advertising; Content match advertising; Display advertising.**

KEYWORDS

internet economics, online advertising, online optimization, duality theory

ACM Reference Format:

Yuan Gao, Kaiyu Yang, Yuanlong Chen, Min Liu, and Noureddine El Karoui. 2022. Bidding Agent Design in the LinkedIn Ad Marketplace. In *Proceedings of AdKDD '22*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

There is a growing trend of automation in online advertising. The manual task of creating tailored ad for each platform and splitting

budget across these platforms are largely automated by programmatic solutions in demand-side platforms (DSPs). The LinkedIn Marketing Solutions is a DSP that allows advertisers to reach LinkedIn members through the LinkedIn Feed as well as its audience network consists of trusted third party publishers. Impression opportunities on these various placements are normally sold through different sorts of auctions. For instance, the LinkedIn Feed placement employs generalized second price (GSP) auctions with reserve price, while majority of publishers in the audience network now uses first price auctions. The paper deals with the design of an automated agent that places bid on behalf of advertisers in these different types of auctions. The goal is to help advertisers maximize their results under natural constraints such as budget delivery plan and expected return on investment (ROI).

Typical advertising objectives include impression, reach, click, social engagement, video view, lead and conversion. Advertisers can directly encode their valuations of the objectives into the bids for each impression opportunity. For example, advertisers who value \$5 per click for their ads could place a bid of $\$5 * p(\text{click})$ in a second price auction, where $p(\text{click})$ is the probability that the user would click after seeing the ad. However this bidding strategy is blind to the budget. This could lead to early termination if the budget is constrained. Throttling-based systems are introduced [2, 23] to address this problem by probabilistically blocking the ad from entering into auctions as a way to smooth out the spend over the entire lifetime. Yet a better strategy in this case is to lower the bids so that budget is spent more effectively.

To formally model this problem, in economics literature [8, 9] a profit-maximization setup is often adopted, where profit is defined as the difference between valuation and payment. In practice, however, valuations from advertisers are not always available. Defining valuations itself is a difficult problem given the scale and complexity of the targeted audience. A preferred way is to frame the bidding problem in a result maximization setup under budget constraint. Valuations from advertisers, if available, can be encoded in this setup via additional constraints on the bids or ROIs.

Mathematically such a bidding problem with budget constraint can be cast in a stochastic optimal control framework [5, 8, 13, 16]. When budget is relatively large compared to the magnitude of bids, fluid approximation [8, 13, 14, 16] is usually applied which greatly simplifies the solution. Explicit bidding formulas are given in [25] under further assumptions on stationary market competition. In practice though, competition in online ad marketplace is highly

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
AdKDD '22, August 15, 2022, Washington, DC, USA

© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00
<https://doi.org/10.1145/1122445.1122456>

dynamic due to the change in demand and supply. For second price auctions, dynamic bid update strategies without stationary assumptions are studied in [9, 19].

Instead of directly optimizing for individual advertiser's interest, there is rich literature dealing with the optimization towards objectives of the platform or DSP, by means of online resource allocations [1, 3, 7, 11, 12, 15, 22]. Bidding strategy for each advertiser in turn arises as a by-product of this optimization. These centralized mechanisms and their equilibrium studies are not considered in our design.

There is limited study on the optimal budget allocation problem across multiple placements. In the case when all placements run second price auctions, a stochastic bandit based solution is proposed [6] where the value for each impression opportunity is assumed to be unknown. This is an overly pessimistic assumption since in reality the value is usually given by a response prediction model. A closely related problem is studied in [24], in which the solution relies on feedback controllers towards pre-calculated ROI target for each placement.

In Section 2 we lay out bidding agent design under a general framework of result maximization under budget constraint. Online methods for solving this optimization problem are discussed in Section 3. Section 4 demonstrates that features such as cost control, budget delivery plan and guaranteed delivery easily translates to constraints in this optimization framework, so that they can be incorporated holistically in one bidding formula.

It's worth noting that the framework is inherently compatible with a collection of ad units and a mixture of different types of selling mechanisms, thereby allowing for the joint optimization of a group of ads across multiple placements each running (possibly) a different auction format. In particular, in Section 5 we show that optimal bidding solution derived from this formulation automatically guarantees the optimality of budget allocation across ad units and placements.

Finally, in Section 6 explicit formulas for the starting bid is derived and various issues around the implementation of online optimization methods are discussed. Experimental results are reported in section 7.

2 GENERAL FORMULATION

In this section we establish a general optimization framework for a single advertiser. For simplicity, the problem is formulated as maximizing total result under a single budget constraint. As noted later in Section 4 it is straightforward to incorporate other types of constraints as well.

Suppose there are a total of T impression opportunities, triggered from targeted users' search or view sessions. The t -th impression has a value of v_t for the agent, which is usually produced from a user response prediction model tied to the advertiser's marketing objective. As an example, for an advertiser optimizing for conversions, v_t represents the impression to conversion rate. Denote $G_t(b)$ as the probability of winning the t -th impression under a bid price of b and $H_t(b)$ as the expected cost. With an advertising budget B the optimization problem becomes

$$\max_{b_t} \sum_{t=1}^T v_t G_t(b_t), \quad s.t. \quad \sum_{t=1}^T H_t(b_t) \leq B. \quad (1)$$

In a complete information setting, we can model $G_t = \mathbf{1}_{[b_t \geq c_t]}$ and $H_t = d_t \mathbf{1}_{[b_t \geq c_t]}$, where c_t is the minimum bid to win impression t , and d_t is the cost if won. This formulation leads to a Knapsack problem [10, 26]. In practice it's often convenient to assume distributional foresight, where G_t and H_t are modeled as smooth functions. Denote their derivatives as g_t and h_t . We make the following assumptions:

- $G_t(0) = 0$, $G_t(+\infty) = 1$ and $g_t > 0$ (monotonicity).
- $H_t(0) = 0$ and $h_t \geq 0$.

These are mild assumptions that generally hold for all practical auction types including first & second price auctions, generalized first price (GFP) auctions, generalized second price (GSP) auctions, Vickrey-Clarke-Groves (VCG) auctions and variants of them with floor price.

The Lagrangian for (1) is

$$L(b_t, \lambda) = \sum_{t=1}^T [v_t G_t(b_t) - \lambda H_t(b_t)] + \lambda B. \quad (2)$$

At fixed $\lambda > 0$, we would choose b_t that maximizes the surplus

$$(v_t/\lambda)G_t(b_t) - H_t(b_t), \quad (3)$$

where v_t/λ is the λ -adjusted value. Optimality condition in b_t implies

$$v_t g_t(b_t^*) = \lambda h_t(b_t^*), \quad (4)$$

or equivalently,

$$b_t^*(\lambda) = \left(\frac{h_t}{g_t}\right)^{-1} \left(\frac{v_t}{\lambda}\right), \quad (5)$$

where $(\cdot)^{-1}$ denotes function inverse.

REMARK 1 (EXPLICIT FORMULA FOR FIRST AND SECOND PRICE AUCTIONS). *In ad marketplace impression opportunities are usually sold via auctions, and H_t and G_t are generally related in various kinds of auctions. In second price auctions, for example, $H_t(b_t) = \int_0^{b_t} b g_t(b) db = b_t G_t(b_t) - \int_0^{b_t} G_t(b) db$. This implies $h_t(b_t) = b_t g_t(b_t)$. According to (4) The optimal bidding strategy in second price auctions is simply bidding the λ -adjusted value*

$$b_t^*(\lambda) = v_t/\lambda.$$

In first price auctions $H_t(b_t) = b_t G_t(b_t)$. The surplus maximization (3) simplifies to $\max_{b_t} (v_t/\lambda - b_t)G_t(b_t)$. Since $h_t(b_t) = G_t(b_t) + b_t g_t(b_t)$, (5) gives the following optimal bidding strategy

$$b_t^*(\lambda) = \left(I + \frac{G_t}{g_t}\right)^{-1} \left(\frac{v_t}{\lambda}\right),$$

where I stands for identity mapping. Note that $G_t/g_t \geq 0$, so we always have $b_t^(\lambda) \leq v_t/\lambda$. This means that in first price auctions the bid has to be shaded on top of the λ -adjusted value.*

We define two useful quantities that will be used throughout.

$$V_t(\lambda) := v_t G_t(b_t^*(\lambda)),$$

the expected value obtained in the t -th opportunity when bidding optimally under λ , and

$$S_t(\lambda) := H_t(b_t^*(\lambda)),$$

the expected spend in the t -th opportunity when bidding optimally under λ . Let $V(\lambda) := \sum_{t=1}^T V_t(\lambda)$ be the total expected value and $S(\lambda) := \sum_{t=1}^T S_t(\lambda)$ the total expected spend.

The following proposition shows that their derivatives are linearly related, and, under mild conditions, both of them are monotonic.

PROPOSITION 1. $V_t'(\lambda) \equiv \lambda S_t'(\lambda), \forall \lambda > 0$. If in addition $\forall h_t > 0$, we have $(\log h_t)' > (\log g_t)'$, then $V_t'(\lambda) \equiv \lambda S_t'(\lambda) \leq 0, \forall \lambda > 0$.

PROOF. The relation $V_t'(\lambda) \equiv \lambda S_t'(\lambda), \forall \lambda > 0$ is directly given by optimality condition (4). Now apply implicit differentiation on (4), we get

$$\frac{db_t^*}{d\lambda} = \frac{h_t(b_t^*)}{v_t g_t'(b_t^*) - \lambda h_t'(b_t^*)}.$$

When $h_t(b_t^*) = 0$ it's clear that $\frac{db_t^*}{d\lambda} = 0$. Otherwise, the condition $(\log h_t)' > (\log g_t)'$ implies

$$\frac{h_t'(b_t^*)}{h_t(b_t^*)} > \frac{g_t'(b_t^*)}{g_t(b_t^*)} = \frac{g_t'(b_t^*)}{(\lambda/v)h_t(b_t^*)} \Rightarrow \frac{db_t^*}{d\lambda} < 0.$$

Finally by chain rule, we have $V_t'(\lambda) = v_t g_t(b_t^*) \frac{db_t^*}{d\lambda} \leq 0$. \square

REMARK 2. In second price auctions, the extra condition $\forall h_t > 0, (\log h_t)' > (\log g_t)'$ in Proposition 1 always holds due to the fact that $h_t(b_t) = b_t g_t(b_t)$. In first price auctions, it's easy to show that the condition translates to log-concavity in G_t , i.e. $G_t G_t'' \leq g_t^2$. Note that G_t can be seen as the cumulative density function (CDF) of competitors' bid distribution, therefore the condition holds as long as the distribution is log-concave.

Plugging $b_t^*(\lambda)$ in the Lagrangian (2), we arrive at the dual problem.

$$\min_{\lambda \geq 0} L(\lambda) := \sum_{t=1}^T \left[V_t(\lambda) - \lambda S_t(\lambda) + \lambda \frac{B}{T} \right]. \quad (6)$$

Setting $L'(\lambda^*) = 0$, and applying Proposition 1 we get

$$\sum_{t=1}^T S_t(\lambda^*) = B. \quad (7)$$

Equation (7) says that an optimal solution $\lambda^* > 0$ would result in a match of the expected spend $S(\lambda)$ with budget. In other words, the solution λ^* is either 0 when the budget is unconstrained, or one that spends the budget, not surprising given Karush–Kuhn–Tucker (KKT) conditions [20].

In addition, Proposition 1 implies the monotonicity of spend $S(\lambda)$. Due to this monotonicity, bisection method can be used to find λ^* given G_t, H_t and v_t for $t = 1, \dots, T$. For example, λ^* hence the optimal bidding sequence b_t can be obtained in hindsight given historical auction logs. However in practice it's difficult to generate accurate forecasts for G_t and H_t in dynamic marketplaces. In the next section we consider online optimization methods in the dual that does not require access to explicit models of these quantities.

3 ONLINE METHODS IN THE DUAL

In this section we present various online optimization methods in the dual (6). Denote the dual loss at impression opportunity t as

$$L_t(\lambda) := V_t(\lambda) - \lambda S_t(\lambda) + \lambda \frac{B}{T}.$$

The dual objective (6) is then $\sum_t L_t(\lambda_t)$. Derivative of the online loss

$$L_t'(\lambda) = B/T - S_t(\lambda),$$

is the difference between the average budget in each opportunity and the expected spend at t .

3.1 Follow the Leader

In follow the leader (FTL) algorithm at each iteration we choose the best solution in hindsight, i.e.,

$$\lambda_{t+1} = \operatorname{argmin}_{\lambda \geq 0} \sum_{\tau=1}^t L_\tau(\lambda).$$

Similar to optimality condition (7) we have

$$\frac{1}{t} \sum_{\tau=1}^t S_\tau(\lambda_{t+1}) = \frac{B}{T}.$$

In practice, this can be achieved by replaying the past auctions to find λ_{t+1} such that the average cost per auction is B/T . As the problem size becomes larger over time, other variants such as those using a fixed size look back window from $t - \Delta t$ to t might be preferred.

In the stationary case (i.e. $G_t = G, H_t = H$), the FTL approach converges to the optimal solution in one iteration (i.e. $\lambda_2 = \dots = \lambda_T = \lambda^*$), but in reality λ_t produced by FTL can be unstable when G_t and H_t are nonstationary.

3.2 Linearization and Follow the Regularized Leader

It's obvious that in both first and second price auctions $L_t(\lambda)$ is convex since $L_t''(\lambda) \geq 0$ according to Proposition 1. Therefore the linearized loss around λ_t

$$\tilde{L}_t(\lambda) := L_t(\lambda_t) + L_t'(\lambda_t)(\lambda - \lambda_t)$$

is an lower bound of $L_t(\lambda)$. In follow the regularized leader (FTRL) we choose

$$\lambda_{t+1} = \operatorname{argmin}_{\lambda \geq 0} \sum_{\tau=1}^t \tilde{L}_\tau(\lambda) + \Omega(\lambda),$$

where $\Omega(\lambda)$ is a regularization term.

THEOREM 3.1 (DUAL ONLINE MIRROR DESCENT). When $\Omega(\lambda) = \lambda^2/(2\epsilon)$, the online update is additive $\lambda_{t+1} = \lambda_t - \epsilon L_t'(\lambda_t)$; when $\Omega(\lambda) = \lambda \ln \lambda/\epsilon$, the online update is multiplicative $\lambda_{t+1} = \lambda_t e^{-\epsilon L_t'(\lambda_t)}$.

Second order approximations to $L(\lambda)$ can be employed to obtain faster convergence. In particular, $L''(\lambda) = -\sum_{t=1}^T S_t'(\lambda) = -S'(\lambda)$ represents the sensitivity of expected spend around λ . Intuitively when the variation of expected spend is small one can make larger updates to λ . Online optimization methods that use second order information, such as natural gradient descent [4] or online Newton's method [18], can be used in the update.

4 OTHER TYPES OF CONSTRAINTS

Apart from the overall budget constraint, in reality there are sometimes preferences on budget delivery schedules as well. Advertisers may also prefer some sort of cost control, e.g., keeping the cost per result under a certain threshold. Other types of constraints include the request to ensure a certain number of results are delivered within a time range. In this section we show it's straightforward to incorporate them in the general formulation.

4.1 Cost Control

A cost per result target C could be specified representing a given ROI goal. The agent attempts to control its cost per result under C by solving the following problem

$$\max_{b_t} \sum_{t=1}^T v_t G_t(b_t), \quad s.t. \quad \sum_{t=1}^T H_t(b_t) \leq \min \left(B, C \sum_{t=1}^T v_t G_t(b_t) \right).$$

Equipping a multiplier for each constraint, the Lagrangian becomes

$$\sum_{t=1}^T [(1 + \mu C)v_t G_t(b_t) - (\lambda + \mu)H_t(b_t)] + \lambda B.$$

Solution in the optimal bidding formula b_t is similar, where

$$b_t^*(\lambda, \mu) = \left(\frac{h_t}{g_t} \right)^{-1} \left(\frac{1 + \mu C}{\lambda + \mu} \cdot v_t \right).$$

Again KKT conditions imply either $\lambda^* = 0$ (in which case the budget is not a binding constraint) or the budget is spent, i.e.,

$$\sum_{t=1}^T H_t(b_t^*(\lambda^*, \mu^*)) = B.$$

Similarly, either $\mu^* = 0$ (in which case the cost per result target is not a binding constraint), or the cost per result is equal to C , i.e.,

$$\sum_{t=1}^T H_t(b_t^*(\lambda^*, \mu^*)) = C \sum_{t=1}^T v_t G_t(b_t^*(\lambda^*, \mu^*)).$$

4.2 Budget Delivery Control

We could define budget delivery constraints on some subintervals $T_k, k = 1, \dots, K$, where $\sum_{t \in T_k} \leq T$. This can reflect advertisers' delivery preferences, for example, to limit the spend on weekends.

$$\max_{b_t} \sum_{t=1}^T v_t G_t(b_t), \quad s.t. \quad \sum_{t=1}^T H_t(b_t) \leq B, \quad \sum_{t \in T_k} H_t(b_t) \leq B_k, \forall k.$$

In practice these constraints can also be dynamic, where they can be added and adjusted in realtime based on inputs from advertisers. The Lagrangian is then

$$\sum_k \sum_{t \in T_k} [v_t G_t(b_t) - (\lambda + \lambda_k)H_t(b_t)] + \sum_k \lambda_k B_k + \lambda B,$$

where $\lambda_k \geq 0$. The new constraints influence the bidding formula. For $t \in T_k$,

$$b_t^*(\lambda, \lambda_k) = \left(\frac{h_t}{g_t} \right)^{-1} \left(\frac{v_t}{\lambda + \lambda_k} \right).$$

Note that each λ_k is only active during the period of T_k , i.e., when the constraint is in place.

4.3 Guaranteed Delivery

Advertisers may also request that a certain number of results be delivered, for instance, during holiday season. Instead of manually increasing the budget during the time period, more precise control can be achieved by encoding the requirement as constraints directly in the optimization framework. More formally, suppose we'd like

V_k number of results to be delivered in each subinterval T_k (defined in the previous section), then the problem becomes

$$\max_{b_t} \sum_{t=1}^T v_t G_t(b_t), \quad s.t. \quad \sum_{t=1}^T H_t(b_t) \leq B, \quad \sum_{t \in T_k} v_t G_t(b_t) \geq V_k, \forall k.$$

The Lagrangian is now

$$\sum_k \sum_{t \in T_k} [(1 + \mu_k)v_t G_t(b_t) - \lambda H_t(b_t)] - \sum_k \mu_k V_k + \lambda B.$$

For $t \in T_k$, the bidding formula is

$$b_t^*(\lambda, \mu_k) = \left(\frac{h_t}{g_t} \right)^{-1} \left(\frac{1 + \mu_k}{\lambda} \cdot v_t \right).$$

Since $\mu_k \geq 0$, a nonzero multiplier μ_k^* would give a boost to the bid during the period T_k to help achieve the delivery requirement.

A combination of the constraints can be implemented at the same time. As an illustration, if all constraints discussed in this section are involved, the bidding formula would then become

$$b_t^*(\lambda, \mu, \lambda_k, \mu_k) = \left(\frac{h_t}{g_t} \right)^{-1} \left(\frac{1 + \mu C + \mu_k}{\lambda + \lambda_k + \mu} \cdot v_t \right).$$

This offers a systematic way to handle a complex set of constraints simultaneously.

5 MULTIPLE PLACEMENTS, GROUP OF ADS AND EQUALITY OF MARGINAL ROI

In advertising, each impression opportunity is associated with a placement. For instance, an advertiser might want to serve his/her ad on multiple publishers at the same time, where each publisher website is a placement. Our framework naturally allows the simultaneous optimization on multiple placements as equation (1) does not differentiate the t -th impression opportunity based on its placement. Concretely, suppose there are a total of K placements and $T_k, k = 1, \dots, K$ represents the set of impression opportunities associated with the k -th placement, with $\sum_{t \in T_k} = T$. Then (1) is essentially

$$\max_{b_t} \sum_{k=1}^K \sum_{t \in T_k} v_t G_t(b_t), \quad s.t. \quad \sum_{k=1}^K \sum_{t \in T_k} H_t(b_t) \leq B.$$

Note that the impression opportunities across different placements can be interleaved in time. According to the Lagrangian (2) the total number of results obtained under optimal bidding is

$$L^* := L(b_t^*, \lambda^*) = \sum_{k=1}^K \left[\lambda^* B_k + \sum_{t \in T_k} V_t(\lambda^*) - \lambda^* S_t(\lambda^*) \right],$$

where B_k is the budget spent on the k -th placement. In this case $\partial L^* / \partial B_k$, the marginal return on investment (ROI) for all placements $1, \dots, K$ are equal (to λ^*), which is a necessary condition for optimal budget allocation across multiple placements.

Similar arguments apply to the scenario of optimizing a group of ads with a global budget. Under the optimal bidding strategy in our framework, budgets are allocated to each ad in an optimal manner to generate the most results. In fact, one major advantage of this framework is its composability. Automated budget allocation across placements, ad units as well as features described in Section 4 are all elegantly handled via a unified bidding strategy.

6 PRACTICAL CONSIDERATIONS

In this section we aim to bridge the gaps between theory and practical implementation. Important topics such as starting bid, forecasting error and various issues around the online updates are discussed. In particular, Section 6.1 gives explicit formulas for the optimal starting bid based on statistics from the targeted audience. Section 6.2 describes several variants of the batch online gradient descent formula and applicability of them in different scenarios.

6.1 Initialization of λ

Section 3 provides incremental update rules for the multiplier λ . The cold start problem, namely the starting value of the multiplier, is also very important as it impacts the time it takes to converge. On the macroscopic level it also influences the price stability of the entire marketplace. Given a set of targeted users, we can learn from these users' past auction logs to initialize λ . In the following theorem we provide explicit formulas for second price auctions.

THEOREM 6.1. *Suppose the competitors' bids follow a log-normal distribution with parameters μ and σ . Additionally, suppose the value of the ad (independently) follows a log-normal distribution with parameters μ' and σ' . Given a budget of B and a total opportunity forecast of T , the λ^* for second price auctions is the solution to*

$$e^{\mu + \frac{\sigma^2}{2}} \Phi \left(\frac{\mu' - \mu - \ln \lambda^* - \sigma^2}{\sqrt{(\sigma')^2 + \sigma^2}} \right) = \frac{B}{T},$$

where Φ is the CDF of a standard normal distribution $\mathcal{N}(0, 1)$.

In practice μ and σ are derived using auction logs from targeted users. The μ' and σ' come from a combination of targeted users' value distribution and ad-specific features. In the multiple placement scenario, the estimation of these quantities can be done at each placement level. The solution λ^* will then be derived based on the traffic forecast T_k from each placement k and a global budget using Theorem 6.1.

6.2 Implementation of Online Methods

Section 3 provides various online update methods that converge to λ^* . In practice we employ a batch version of the dual online mirror descent algorithm, where the multiplier λ_t is updated every time interval dt . In the following we focus on the additive formula in Theorem 3.1 though the arguments apply to the multiplicative update as well. Let N_{dt} be the number of impression opportunities in time dt , then the (batch) online update becomes

$$\lambda_{t+N_{dt}} = \lambda_t - \epsilon \left(\frac{B}{T} \cdot N_{dt} - S_{dt} \right), \quad (8)$$

where S_{dt} is the sum of expected spend during the past update period under λ_t . This can simply be set as the observed spend in dt . However if the charge events are sparse, some estimator based on the observed spend is needed to reduce the variance. (8) is equivalent to

$$\lambda_{t+N_{dt}} = \lambda_t - \epsilon_{dt} (1 - R_{dt}), \quad (9)$$

where

$$\epsilon_{dt} := \epsilon \frac{BN_{dt}}{T}$$

is the step size, and

$$R_{dt} := \frac{S_{dt}/N_{dt}}{B/T} \quad (10)$$

is the ratio of average cost per opportunity and average budget per opportunity.

The choice of dt is important as it controls the tradeoff between frequency of update and variance in S_{dt} . An alternative is to keep track of the observed impression opportunities and to trigger an update after a fixed amount of observations.

6.2.1 Normalization. The scale of λ varies a lot across advertisers, due to the diversity in budget and targeting setups. We therefore choose a normalizing constant λ' for each advertiser and define $\tilde{\lambda}_t = \lambda_t/\lambda'$ to make the update dimensionless:

$$\tilde{\lambda}_{t+N_{dt}} = \tilde{\lambda}_t - \eta_{dt} (1 - R_{dt}),$$

where

$$\eta_{dt} := \epsilon \frac{BN_{dt}}{\lambda' T} \quad (11)$$

becomes a dimensionless step size that is easier to tune in practice. However, the choice of λ' for each advertiser plays an important role in convergence if a global parameter $\xi := \epsilon B/\lambda'$ is selected such that $\eta_{dt} = \xi N_{dt}/T$. To see that, note that the inverse of λ_t will be used to compute the bids in auctions, and

$$\frac{1}{\lambda_{t+N_{dt}}} = \frac{1}{\lambda'(\tilde{\lambda}_t - \eta_{dt}(1 - R_{dt}))} = \frac{1/\lambda_t}{1 - (\xi N_{dt}/T)(1 - R_{dt})(\lambda'/\lambda_t)}.$$

Therefore a $\lambda' < \lambda^*$ would slow down the convergence whereas a $\lambda' > \lambda^*$ tends to create oscillatory behavior asymptotically due to magnification of the noise in $1 - R_{dt}$. In practice we find setting the normalization factor as the initialization provided in Section 6.1 works well.

6.2.2 Total Forecast vs. Relative Forecast. Sometimes it is easier to forecast the relative traffic pattern over time rather than the absolute number of opportunities T . In that case one can use \hat{r}_{dt} , the forecasted proportion of traffic in time dt , to replace N_{dt} and T in the formula of R_{dt} in (10) and η_{dt} in (11). The formula then becomes $R_{dt} = S_{dt}/(B\hat{r}_{dt})$ and step size $\eta_{dt} = \epsilon B\hat{r}_{dt}/\lambda'$. The estimator \hat{r}_{dt} can also be dynamically adjusted based on past observed traffic N_{dt} .

6.2.3 Model Predictive Control. One can reset the average budget per opportunity target B/T at each update. A model predictive control (MPC) version of the update would set R_{dt} as

$$\frac{S_{dt}/N_{dt}}{(B - B_{t+N_{dt}})/(T - t - N_{dt})}. \quad (12)$$

This version of the update would encourage budget exhaustion near the end of the ad's lifetime.

6.2.4 Empirical vs. Asymptotic Regret. In terms of convergence of online optimization methods, the asymptotic regret has been the major object studied in the literature [17]. However due to the fact that ad campaigns have a limited lifetime, the number of λ updates in (8) for each campaign is limited. In practice, we find it favorable to measure the empirical regret instead, and to use that as a metric in the selection of different variants of optimization methods and the hyperparameters within them.

7 EXPERIMENTS

7.1 Offline Experiments

7.1.1 Bid Initialization. To evaluate the initialization method proposed in 6.1, we sampled $T = 100,000$ auctions from the internal log. Each auction is associated with a single impression where the winner is charged by the supporting second bid. Based on this dataset, we can infer the optimal bid for an ad under a hypothetical budget, if it were to participate in all these auctions. For simplicity we assume the ad under consideration is a brand ad maximizing for impressions, therefore we have $v_t \equiv 1$ in (1), and according to (5) the optimal bidding strategy is to bid $1/\lambda^*$ throughout. Figure 1 shows the change of optimal bid under various budget levels. Alongside we also plot the percent error of the solution produced by 6.1, where parameters μ and σ of the log-normal distribution are fit using the winners' bids in the sampled auctions.

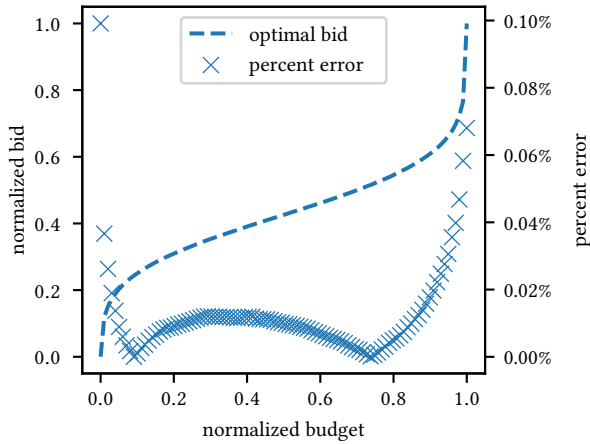


Figure 1: Optimal bid as a function of budget. Percent error of the solution produced by 6.1 compared to the optimal solution is also shown. Both the budget and optimal bid are rescaled to $[0, 1]$.

7.1.2 Dual Online Mirror Descent. Using the same auction dataset, we run the dual online mirror descent algorithm specified in equation (9) where ϵ_{dt} is simply set to 1. Again for simplicity we assume the optimization for a brand ad where $v_t \equiv 1$, so that the bid submitted in each auction t is $1/\lambda_t$, where λ_t is updated using equation (9). The budget of the ad is set as 1% of the total cost of impressions in the auction dataset.

Two variants of the algorithm are considered – the *vanilla* version where R_{dt} is defined by equation (10) and the *MPC* version where R_{dt} is defined by equation (12). For both variants, we trigger a bid update every 1000 auctions (i.e., $N_{dt} \equiv 1000$). Figure 2 shows the trajectory of bid under the two variants, assuming a starting bid that is lower than the optimal bid $1/\lambda^*$ in hindsight. Given the nonstationarity of the auction price distribution, the algorithm never converges to a stationary bid. However the vanilla variant pushes the bid close to the optimal bid after some iterations and the bid oscillates around it afterwards. The MPC variant exhibits

similar behavior yet it maintains a higher bid in the later stage, to compensate for the loss of opportunities due to a low initial bid.

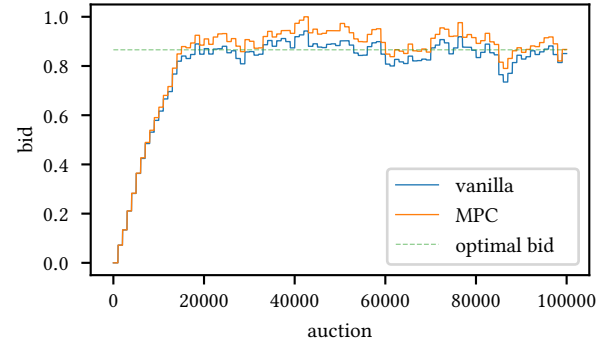


Figure 2: Sample path of bid for two algorithm variants. Bid is rescaled to $[0, 1]$.

Finally, we experiment with varying starting bids for the MPC variant. The motivation is that in practice it's challenging to get an accurate estimation of the optimal bid. The errors of bid initialization shown in Figure 1 is small because it's evaluated on the training dataset, and the errors generally get magnified when there is forecast error in quantities such as μ, σ and T .

As illustrated in Figure 3, the MPC variant is robust to the starting point in terms of budget delivery. However the cost per impression (CPI), a measure of advertiser ROI, can be severely impacted when the starting bid is away from the optimal bid. In particular, the impact is not symmetric with respect to the deviation. Overestimated starting bids incur greater harm to CPI due to early spending at high bid.

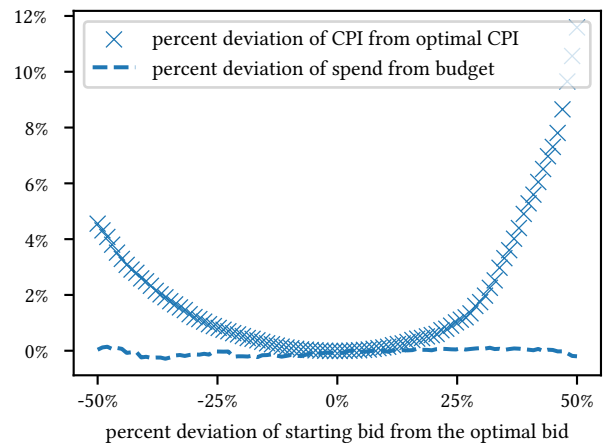


Figure 3: Performance of the MPC variant under different starting bids.

7.2 Online Results

Bidding agent based on this framework was implemented on the LinkedIn ad marketplace and is compared with a prior feedback-control agent which adjusts the bid to track the forecast traffic

curve. For unbiased evaluation the two agents are compared using the budget-split experimentation platform [21]. We observe a statistically significant increase of 8.25% in advertiser ROI (with neutral platform revenue) using the new method.

ACKNOWLEDGMENTS

We would like to thank Yi Zhang and Onkar Dalal for their support. We appreciate Qian Yao and Ricardo Salmon for helpful discussions.

REFERENCES

- [1] Zoë Abrams, S Sathya Keerthi, Ofer Mendelevitch, and John A Tomlin. 2008. Ad delivery with budgeted advertisers: A comprehensive LP approach. *Journal of Electronic Commerce Research* 9, 1 (2008).
- [2] Deepak Agarwal, Souvik Ghosh, Kai Wei, and Siyu You. 2014. Budget pacing for targeted online advertisements at linkedin. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1613–1619.
- [3] Gagan Aggarwal, Ashwinkumar Badanidiyuru, and Aranyak Mehta. 2019. Autobidding with constraints. In *International Conference on Web and Internet Economics*. Springer, 17–30.
- [4] Shun-ichi Amari, Hyeyoung Park, and Kenji Fukumizu. 2000. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural computation* 12, 6 (2000), 1399–1409.
- [5] Kareem Amin, Michael Kearns, Peter Key, and Anton Schwaighofer. 2012. Budget optimization for sponsored search: Censored learning in MDPs. *arXiv preprint arXiv:1210.4847* (2012).
- [6] Vashist Avadhanula, Riccardo Colini Baldeschi, Stefano Leonardi, Karthik Abinav Sankararaman, and Okke Schrijvers. 2021. Stochastic bandits for multi-platform budget optimization in online advertising. In *Proceedings of the Web Conference 2021*. 2805–2817.
- [7] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. 2020. The best of many worlds: Dual mirror descent for online allocation problems. *arXiv preprint arXiv:2011.10124* (2020).
- [8] Santiago R Balseiro, Omar Besbes, and Gabriel Y Weintraub. 2015. Repeated auctions with budgets in ad exchanges: Approximations and design. *Management Science* 61, 4 (2015), 864–884.
- [9] Santiago R Balseiro and Yonatan Gur. 2019. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science* 65, 9 (2019), 3952–3968.
- [10] Deeparnab Chakrabarty, Yunhong Zhou, and Rajan Lukose. 2008. Online knapsack problems. In *Workshop on internet and network economics (WINE)*.
- [11] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. 2011. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1307–1315.
- [12] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S Mirrokni, and Cliff Stein. 2010. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*. Springer, 182–194.
- [13] Joaquin Fernandez-Tapia, Olivier Guéant, and Jean-Michel Lasry. 2017. Optimal real-time bidding strategies. *Applied Mathematics Research eXpress* 2017, 1 (2017), 142–183.
- [14] Guillermo Gallego and Garrett Van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science* 40, 8 (1994), 999–1020.
- [15] Paul Grigas, Alfonso Lobos, Zheng Wen, and Kuang-chih Lee. 2017. Profit maximization for online advertising demand-side platforms. In *Proceedings of the ADKDD '17*. 1–7.
- [16] Ramki Gummadi, Peter Key, and Alexandre Proutiere. 2013. Optimal bidding strategies and equilibria in dynamic auctions with budget constraints. *Available at SSRN 2066175* (2013).
- [17] Elad Hazan. 2019. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207* (2019).
- [18] Elad Hazan, Amit Agarwal, and Satyen Kale. 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69, 2-3 (2007), 169–192.
- [19] Niklas Karlsson. 2020. Feedback Control in Programmatic Advertising: The Frontier of Optimization in Real-Time Bidding. *IEEE Control Systems Magazine* 40, 5 (2020), 40–77.
- [20] Harold W Kuhn and Albert W Tucker. 2014. Nonlinear programming. In *Traces and emergence of nonlinear programming*. Springer, 247–258.
- [21] Min Liu, Jialiang Mao, and Kang Kang. 2020. Trustworthy online marketplace experimentation with budget-split design. *arXiv preprint arXiv:2012.08724* (2020).
- [22] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2007. Adwords and generalized online matching. *Journal of the ACM (JACM)* 54, 5 (2007), 22–es.

- [23] Jian Xu, Kuang-chih Lee, Wentong Li, Hang Qi, and Quan Lu. 2015. Smart pacing for effective online ad campaign optimization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2217–2226.
- [24] 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.
- [25] 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.
- [26] Yunhong Zhou and Victor Naroditskiy. 2008. Algorithm for stochastic multiple-choice knapsack problem and application to keywords bidding. In *Proceedings of the 17th international conference on world wide web*. 1175–1176.

A PROOF OF THEOREM 3.1

This is standard result in online optimization [17]. Ignoring the constant terms, we have

$$\lambda_{t+1} = \operatorname{argmin}_{\lambda \geq 0} \sum_{s=1}^t L'_s(\lambda_s) \cdot \lambda + R(\lambda).$$

First order optimality condition gives $R'(\lambda_{t+1}) = -\sum_{s=1}^t L'_s(\lambda_s)$. In case $R(\lambda) = \lambda^2/(2\epsilon) \Rightarrow R'(\lambda) = \lambda/\epsilon$, then

$$\lambda_{t+1} = -\epsilon \left(\sum_{s=1}^{t-1} L'_s(\lambda_s) + L'_t(\lambda_t) \right) = -\epsilon \left(\frac{\lambda_t}{-\epsilon} + L'_t(\lambda_t) \right) = \lambda_t - \epsilon L'_t(\lambda_t).$$

Similarly, $R(\lambda) = \lambda \ln \lambda/\epsilon \Rightarrow R'(\lambda) = (\ln \lambda + 1)/\epsilon$, then

$$\lambda_{t+1} = e^{-1-\epsilon \sum_{s=1}^{t-1} L'_s(\lambda_s) - \epsilon L'_t(\lambda_t)} = \lambda_t e^{-\epsilon L'_t(\lambda_t)}.$$

B PROOF OF THEOREM 6.1

The following lemma is useful in the proof. Let ϕ and Φ be the PDF and CDF of a standard normal distribution $\mathcal{N}(0, 1)$.

LEMMA B.1. *Suppose $X \sim \mathcal{N}(0, 1)$, then $E[\Phi(aX+b)] = \Phi\left(\frac{b}{\sqrt{1+a^2}}\right)$.*

PROOF.

$$\begin{aligned} E[\Phi(aX+b)] &= \int_{-\infty}^{\infty} \int_{-\infty}^{ax+b} \phi(y) dy \phi(x) dx \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{1}_{[y \leq ax+b]} \cdot \phi(y) \phi(x) dy dx. \end{aligned}$$

This is simply the probability of $Y \leq aX + b$ given that X, Y are i.i.d standard normally distributed variables. Since $Y - aX \sim \mathcal{N}(0, 1+a^2)$, this completes the proof. \square

Now we proceed with the proof of the theorem.

PROOF. Denote $p_{\mu, \sigma}(x)$ as the probability density function for *Lognormal*(μ, σ). The expected cost per opportunity under a given λ is

$$S(\lambda) := \int_0^{\infty} \left(\int_0^{v/\lambda} z p_{\mu, \sigma}(z) dz \right) p_{\mu', \sigma'}(v) dv,$$

where at each given value v we would bid v/λ (see Remark 1). First note that the inner integral $\int_0^{v/\lambda} z p_{\mu, \sigma}(z) dz$ is the partial expectation of a log-normal random variable. By standard results

$$\int_0^{v/\lambda} z p_{\mu, \sigma}(z) dz = e^{\mu + \frac{\sigma^2}{2}} \Phi\left(\frac{\ln v - \ln \lambda - \mu - \sigma^2}{\sigma}\right).$$

That implies $S(\lambda) = e^{\mu + \frac{\sigma^2}{2}} E \left[\Phi \left(\frac{\ln V - \ln \lambda - \mu - \sigma^2}{\sigma} \right) \right]$, where V is r.v. with $\text{Lognormal}(\mu', \sigma')$. Now with a change of variable $X := \frac{\ln V - \mu'}{\sigma'} \sim \mathcal{N}(0, 1)$, we have

$$S(\lambda) = e^{\mu + \frac{\sigma^2}{2}} E \left[\Phi \left(\frac{\sigma' X + \mu' - \mu - \ln \lambda - \sigma^2}{\sigma} \right) \right].$$

Now apply the results from Lemma B.1 ,

$$S(\lambda) = e^{\mu + \frac{\sigma^2}{2}} \Phi \left(\frac{\mu' - \mu - \ln \lambda - \sigma^2}{\sqrt{(\sigma')^2 + \sigma^2}} \right).$$

□