A Field Guide for Pacing Budget and ROS Constraints

Santiago R. Balseiro [1 2] **Kshipra Bhawalkar** [1] **Zhe Feng** [1] **Haihao Lu** [1] **Vahab Mirrokni** [1]
Balasubramanian Sivan [1] **Di Wang** [1]

Abstract

Budget pacing is a popular service that has been offered by major internet advertising platforms since their inception. In the past few years, autobidding products that provide real-time bidding as a service to advertisers have seen a prominent rise in adoption. A popular autobidding stategy is value maximization subject to return-on-spend (ROS) constraints. For historical or business reasons, the systems that govern these two services, namely budget pacing and ROS pacing, are not necessarily always a single unified and coordinated entity that optimizes a global objective subject to both constraints. The purpose of this work is to theoretically and empirically compare algorithms with different degrees of coordination between these two pacing systems. In particular, we compare (a) a fully-decoupled *sequential algorithm*; (b) a minimally-coupled *min-pacing algorithm*; (c) a *fully-coupled* dual-based algorithm. Our main contribution is to theoretically analyze the min-pacing algorithm and show that it attains similar guarantees to the fully-coupled canonical dual-based algorithm. On the other hand, we show that the sequential algorithm, even though appealing by virtue of being fully decoupled, could badly violate the constraints. We validate our theoretical findings empirically by showing that the min-pacing algorithm performs almost as well as the canonical dual-based algorithm on a semi-synthetic dataset that was generated from a large online advertising platform's auction data.

1. Introduction

Internet advertisers purchase advertising opportunities by bidding in real-time auctions, and, to control their expenditures, it is common for advertisers to set budgets for their campaigns (Google, 2023b; Meta, 2023; Twitter, 2023).

[1]Google Research, USA [2]Columbia University, NYC, USA. Correspondence to: Di Wang <wadi@google.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Budget pacing is a popular service offered by most advertising platforms that allows advertisers to specify their budgets and then optimizes advertiser bids in real-time to maximize advertisers' return subject to the spend being at most the budget. In the past few years, thanks to the increasing availability of ROS-related metrics, and the vastly improved conversion prediction models, autobidding products have seen a prominent rise in adoption (Center, [n. d.]; Google, 2023a). These tools provide value-optimizing real-time bidding subject to return-on-spend (ROS) constraints (on top of the existing budget constraints). Autobidding takes as input high-level advertiser goals like the target cost per conversion or acquisition and places real-time bids on a per-query basis to optimize advertiser returns.

The algorithms that govern budget and ROS pacing, namely value-optimization subject to budget and ROS constraints, are not necessarily always a unified entity that optimizes a global objective. See Appendix B for a detailed discussion on the reasons. These services are often managed by different business units within the same organization or by different organizations altogether (many third-party demand-side platforms offer autobidding services). This results in different algorithms independently choosing/modifying advertisers' bids. This is not surprising in light of the meaningful gap between the times at which these products gained traction, with budget pacing systems having been standard and popular much earlier. As a result, even if the objectives of both services are aligned, the presence of budget and ROS constraints can introduce inefficiencies in the bidding process when the systems are decoupled. How do the fully decoupled and fully coupled optimal pacing services compare? Is there a way to operate the pacing service that obtains the best of both worlds: i.e., (a) maintain the theoretical guarantees of the fully coupled optimal pacing service, while (b) still being only minimally coupled? Our contribution in this work is to design and analyze an algorithm that approaches the best of both worlds. We establish this fact both theoretically and empirically.

1.1. Pacing Services

Pacing services are online algorithms that adaptively adjust advertisers' bids based on auction feedback to maximize certain objectives while satisfying different constraint.

Nowadays, a popular paradigm in internet advertising markets is that of *value maximization* (Center, [n. d.]; Google, 2023a). Unlike the usual quasilinear utility model, where the bidder seeks to maximize the difference between their value and payment, the bidder's stated objective in autobidding/budgeting products is to maximize their overall value (e.g., the number of conversions or conversion value) while respecting their budget and ROS constraints. For example, a bidder could ask to maximize the total number of conversions they get, subject to spending at most $1000 and not paying more than $5 per conversion. Figure 1a illustrates a *joint optimization pacing service*, which we also refer to as a *dual-optimal pacing service*, which takes as input the advertiser's budget and ROS target, and then automatically bids on behalf of the advertiser in the platform's auction. Importantly, the pacing services maintain a feedback loop that monitors the real-time spend and conversions from the auction and uses this information to adjust bids.

In Figure 1b we illustrate a typical *sequential pacing service* in which the ROS pacing service feeds bids to the budget pacing service, which, in turn, bids in the platform's auction. Each service consumes the spend and conversion feedback from the auction to adjust bids dynamically. The benefit of the sequential optimization architecture is its decoupled nature, i.e., it can operate separate modules for budget pacing and ROS pacing.

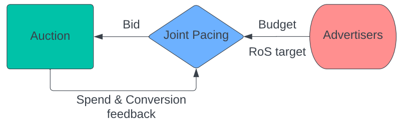
We also consider a third minimally coupled architecture (Figure 1c), which we call the *min pacing service*. Rather than organizing the pacing services sequentially, they are organized in parallel. For each auction, the bid is obtained by taking the minimum of the bids generated by the two systems. While more generally one can think about other reduction operations of the two pacing systems' bids, as we show in this work, the min pacing already performs quite well and achieves the performance of the joint dual-optimal pacing service both theoretically and empirically, while still being only minimally coupled. Moreover, a significant advantage of the min pacing service (due to its minimally coupled nature) is its practicality, whereas the joint dual-optimal pacing, although being the natural solution following standard duality, can be prohibitive to implement in practice.

To see why a fully-coupled pacing system can be less feasible in practice, we start with how these services are provided in the broader context of online advertising. In the industry, budget constraints and ROS constraints are handled by separate teams for various reasons. For clarity, we'll refer to the former service as budget pacing and the latter as auto-bidding. In our optimization formulation, we assume the value is given, but in practice, it is usually a predicted quantity (e.g., predicted click-through or conversion rate). One main task of the auto-bidding service, on top of setting the bids, is actually to come up with the predictions. There is great heterogeneity in the prediction task across different types of advertising campaigns based on: (1) what features are available for prediction, e.g. campaigns from different channels such as search, video, news feed, app store, display or cross-channel; (2) the value that's actually getting predicted, e.g. service sign-up, in-app purchase, unique viewership etc. Thus, the (in-house) auto-bidding service is provided by different teams (across different organizations) for different types of campaigns. On the other hand, budget pacing has existed since the very early days and is fairly homogeneous for all types of campaigns, so this service is typically provided by one single team for all campaigns. Advertisers usually use budget pacing provided by the platform since it has the spending information with the lowest latency (to avoid overspending). Consequently, auto-bidding and budget pacing are run in different servers with different underlying infrastructure, data pipeline, logging, and latency requirements, all in a distributed manner at the internet scale. We include more background on the practicality considerations in Appendix B.

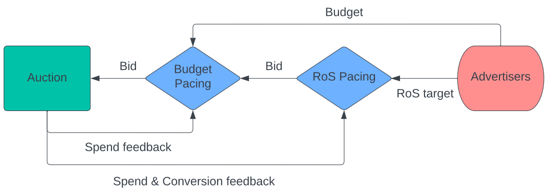
1.2. Our Results

We compare all three algorithms described above, both theoretically and empirically. We next describe the algorithmic implementations of the pacing services, the empirical evaluation, and our theoretical analysis. Our main contribution is a theoretical analysis of the min-pacing algorithm and shows that it obtains the best of both worlds, i.e., its performance approaches that of the joint dual-optimal pacing service, while still being essentially decoupled much like the sequential pacing architecture. On the other hand, we show that the sequential architecture itself is a very poor choice: it either violates constraints by $\Omega(T)$ or has a regret of $O(T)$, when there is a finite horizon of T repeated auctions.

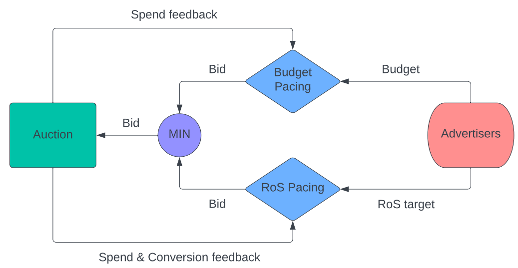
Algorithmic implementation. In this work, we consider *uniform bidding policies* (which were first proposed and analyzed in Feldman et al. 2007) that multiplicatively scale advertisers' values, which are usually generated using advanced machine learning prediction algorithms (McMahan et al., 2013; He et al., 2014; Zhou et al., 2018; Juan et al., 2016; Lu et al., 2017). Uniform bidding is appealing for its simplicity, can be shown to be optimal in many settings, and is extensively used in practice (Aggarwal et al., 2019). The bid multiplier k of the uniform bidding policy is adjusted in real-time using a feedback loop. While many choices are possible for the feedback loop, in this work we consider Lagrangian dual algorithms, which are the work-horse algorithms of budget pacing (Balseiro and Mirrokni, 2022). At a high level, these algorithms introduce a dual variable for each constraint and then adjust these dual variables dynamically using a first-order algorithm. The final bid multiplier is calculated using these dual variables. Dual-based algo-



(a) Dual-Optimal Pacing Service



(b) Sequential Pacing Service



(c) Min Pacing Service

Figure 1. Three Different Pacing Services for Budget and ROS Constraints

	ROS Violation	Regret
Dual-Optimal	$O(\sqrt{T})$	$O(\sqrt{T})$
Sequential	$\Omega(T)$	$\Omega(T)$
Min	$O(\sqrt{T})$	$O(\sqrt{T})$

Table 1. Summary of our theoretical evaluation.

rithms have strong performance guarantees and have been shown to subsume PID controllers—one of the most popular feedback controllers used in practice (Tashman et al., 2020; Zhang et al., 2016; Smirnov et al., 2016; Yang et al., 2019; Ye et al., 2020; Balseiro et al., 2022c). Therefore, we believe the algorithms studied in this paper are representative of those used by pacing services in practice. We provide more details on the concrete algorithmic implementation in Section 2.

Theoretical evaluation. We evaluate the three algorithms along two dimensions: ROS constraint error, and conversion value. Budget constraints are hard in practice, i.e., pacing algorithms can no longer participate in auctions when budgets are exhausted. In contrast, ROS constraints are often soft: while small violations are permitted, large violations are undesirable. Finally, the conversion value garnered before the budget runs out should be as large as possible. We benchmark algorithms by looking at their regret relative to the conversion value of an offline optimum pacing strategy satisfying budget and ROS constraints. Our results are summarized in Table 1.

Technical contribution. Our evaluation is performed in a statistical environment under uncertainty. In other words, we assume that values and competing bids are drawn independently from a distribution that is unknown to the algorithms. We consider a finite horizon with T repeated auctions in which the budget B is proportional to the number of auctions, i.e., $B = \rho T$ for some fixed $\rho > 0$. Recently, (Feng et al., 2022) showed that the joint dual-optimal algorithm scores high along all dimensions. It violates the ROS constraint by an amount $O(\sqrt{T})$, and attains a regret (conversion value garnered before the budget runs out relative to offline optimum) of $O(\sqrt{T})$. Our main result in this paper is to show that the min pacing algorithm also scores high along both dimensions, achieving $O(\sqrt{T})$ bounds similar to those of the dual-optimal pacing algorithm. The analysis of the min pacing algorithm is challenging because we do not have access to a Lyapunov function, as in the dual-optimal pacing case. Instead, we analyze the algorithm by carefully studying the dynamics of the dual variables, which evolve according to a complex stochastic process. In particular, using the ODE technique for recursive algorithms, we first prove that the min-pacing algorithm quickly identifies which constraint binds and reaches the orbit of an optimal solution in $O(\sqrt{T})$ time steps. Then, using stochastic stability tools, we argue that the algorithm never leaves the orbit of an optimal solution with high probability. We conclude by showing that the regret accumulated once the algorithm is in orbit is small using results from online convex optimization.

We finally argue that sequential pacing leads to unacceptable levels of ROS constraint violation or regret. In particular, we show any instantiation of the sequential pacing algorithm can have either linear (i.e. $\Omega(T)$) ROS violation or linear regret on some instances.

Empirical evaluation. Section 4 explains in detail our evaluation methodology, including how we construct our semi-synthetic dataset, how we obtain the different quantities in our optimization formulation (1) based on real auction data. Here we give a high-level summary of our result. The objective of the algorithms is to maximize conversion value subject to budget and ROS constraints. In our simulations, as explained in Section 4, we strictly enforce the budget constraint, terminating simulations once it's violated. However, we do not impose a similar hard limit on the ROS constraint. This aligns with practice, as budget constraints are typically more rigid than ROS constraints. Consequently, direct comparison of conversion values is not possible, as some algorithms might produce solutions that violate the ROS constraint. Therefore, we evaluate the different algorithms as follows. For each algorithm, we determine for each percentile level $z\%$ violation of the ROS constraint, the total conversion value obtained by the algorithm over all the campaigns that violated the constraint by at most $z\%$. By comparing these quantities, we can obtain the following critical insight: what percentage of ROS constraint violation does the naive sequential pacing need, to obtain the same value as the dual-optimal pacing does, at say 1% constraint violation, or the min pacing does, at say 5% constraint violation. Such plot is shown in Figure 2b. A similar plot, but instead focusing on the number of campaigns that violate the ROS constraint by $z\%$ is portrayed in Figure 2a.

The high level summary is quite evident from these figures: *the naive sequential pacing needs to violate the ROS constraint by a very significant percentage to approach anywhere near the dual-optimal pacing, while the min pacing approaches the dual optimal pacing at a much smaller percentage of ROS constraint violation. Moreover, in sequential pacing, the feedback loops of budget and ROS can lead to unstable dynamics.* Our findings suggest avoiding the sequential implementation despite its simplicity and appeal, and point towards having the two feedback loops either operating in a centralized manner, or at least minimally coupled as in the min pacing architecture. Overall, our work has implications for the design and operation of pacing services. Our findings suggest that the lack of coordination of sequential pacing can lead to suboptimal and unstable outcomes. Advertising should, whenever possible, adopt algorithms that have some level of coordination between budget and ROS pacing. If centralized architecture is not an option, then the minimally-coupled min pacing architecture is a simple, practical and high-performant option to consider.

1.3. Related Work

We discuss here the paper that is most related to ours, and due to lack of space, we discuss other related work in Appendix A. In independent work, Lucier et al. (Lucier et al., 2023) studied a conceptually similar, yet different, algorithm that uses the final bid as the minimum of the bids from the two pacing services. But unlike ours, the bid they use from each pacing service is different from the dual-optimal bid for that service. More importantly, they study a multi-bidder setting (unlike our single bidder setting) and their primary quantity of interest is the loss in liquid welfare, namely, the budget-capped sum of values obtained by all agents, when all of them employ this bidding algorithm. They establish that when the autobidding algorithms of agents play against each other, the resulting expected liquid welfare is at least half of the optimal expected liquid welfare achievable. For a single-bidder setting in which competing bids are drawn i.i.d. their results imply a regret bound of $O(T^{7/8})$ as opposed to the tight $O(\sqrt{T})$ guarantee we prove.

2. The Setup

In this section, we define a formal model for budget and ROS constraint pacing. We consider a single bidder who participates in T repeated auctions. The bidder derives a value of $v_t \in [0, 1]$ from getting allocated in auction $t = 1, \ldots, T$. Upon submitting a bid of b_t, the bidder gets an allocation of $x_t(b_t)$ and an expected payment of $p_t(b_t)$. I.e., $x_t : \mathcal{R}_{\geq 0} \to [0, 1]$, and $p_t : \mathcal{R}_{\geq 0} \to [0, 1]$ are the allocation and payment functions respectively. Note we assume without loss of generality (by scaling) that v_t, x_t, p_t are all in $[0, 1]$. The tuple $\gamma_t = (v_t, x_t, p_t)$ is drawn i.i.d. every round from an unknown distribution $\mathcal{P}$. We denote the sequence of T samples by $\overrightarrow{\gamma} := \{\gamma_1, \gamma_2, \ldots, \gamma_T\} \sim \mathcal{P}^T$ and sequences of length $\ell \neq T$ by $\overrightarrow{\gamma}_\ell$ where needed. At the beginning of round t, the bidder has knowledge of the value v_t and the historical information of past auctions to decide on a bid, b_t. Denote $\delta_t = (x_t(b_t), p_t(b_t))$ to represent the outcome of the auction at round t. At the end of round t, the bidder observes δ_t. Thus, the historical information at the beginning of round t is $h_t = \{(v_s, \delta_s)\}_{s \leq t-1}$.

The optimization objective. The advertiser is a *value-maximizer* and seeks to maximize the overall value while respecting the budget of B dollars and the ROS constraint. Formally, the bidder's optimization problem is stated as follows:

$$\begin{array}{ll} \underset{b_t: t=1,\cdots,T}{\text{maximize}} & \sum_{t=1}^T v_t \cdot x_t(b_t) \\ \text{subject to} & \sum_{t=1}^T p_t(b_t) \leq \sum_{t=1}^T v_t \cdot x_t(b_t), \\ & \sum_{t=1}^T p_t(b_t) \leq B \end{array} \tag{1}$$

The first constraint is the ROS constraint, which states that for every dollar spent, there is at least a dollar of value.[1] The second constraint is the budget constraint. We define the per-round budget by $\rho := B/T$. In round t the bidder bids $b_t = \pi_t(v_t, h_t)$. The function $\pi_t(\cdot, \cdot)$ could be randomized.

Truthful auctions, nontruthful auctions, uniform bid-

[1]More generally, one can have the constraint to state that for every dollar spent, there is at least τ dollars of value. But without loss of generality, one can set $\tau = 1$. The update to the bidding formula as a function of τ is quite straightforward, and we skip this here to avoid carrying the notational clutter of τ everywhere.

ding policy. We restrict attention to a uniform bidding policy, which computes a bid multiplier k_t independently of the current value v_t so that the bid submitted is $b_t = k_t \cdot v_t$. If the underlying auction is truthful[2], Aggarwal et al. (2019) showed that the optimal bidding algorithm for problem (1) is indeed a uniform bidding policy, and hence the restriction to uniform bidding is without loss of generality. If the underlying auction is non-truthful, the restriction to uniform bidding can be made without loss if the buyer has access to an optimizer $g_t(v)$ that computes the optimal bid to submit in a one-shot auction for any given true value[3] v. In this case, bidding $b_t = g_t(k_t \cdot v)$ would be optimal for the bidder due to the revelation principle.

2.1. The Bidding Algorithms

Despite the simplicity and appeal of uniform bidding, computing the optimal multiplier k_t requires knowledge of the sample path $\overrightarrow{\gamma}$, while information is only revealed online. Thus, to approach the performance of uniform bidding policy in an online setting, a standard technique is to dualize the constraints and look at the Lagrangian dual of the problem. We introduce dual variables $\mu \geq 0$ for the budget constraint and λ for the ROS constraint and write Lagrangian dual of the problem (1):

$$\min_{\lambda\geq 0,\mu\geq 0} \quad \max_{b_t:t=1,\cdots,T} \left\{ T\rho\mu + \sum_{t=1}^{T}\Big((1+\lambda)v_t\cdot x_t(b_t) - (\mu+\lambda)p_t(b_t)\Big)\right\} . \tag{2}$$

At each time t, the Lagrangian dual variables λ_t, μ_t can be updated using first-order algorithms, which, as we explain below, employ as gradients the per-period constraint violation. Then, we compute the multiplier k_t as a function of λ_t, μ_t and set the bid of $b_t = k_t \cdot v_t$.

Dual-Optimal Pacing. We now discuss how to derive the optimal bidding multiplier k_t when the underlying auction is truthful. Note that the Lagrangian dual problem (2) becomes separable over time after dualizing the constraints. Therefore, at time t, assuming that the dual variables are μ_t and λ_t, the optimal bid by solving (2) is

$$\begin{aligned} b_t^{\text{dual-opt}} &= \operatorname*{argmax}_b \{(1+\lambda_t)\cdot v_t\cdot x_t(b) - (\mu_t+\lambda_t)p_t(b)\} \\ &= \operatorname*{argmax}_b \left\{\frac{1+\lambda_t}{\mu_t+\lambda_t}\cdot v_t\cdot x_t(b) - p_t(b)\right\} = \frac{1+\lambda_t}{\mu_t+\lambda_t}\cdot v_t\,, \end{aligned} \tag{3}$$

where the second equation follows from extracting the factor $(\mu_t+\lambda_t)$ and the last because the bidder's problem is equivalent to that of bidding in a truthful auction when the value is $(1+\lambda_t)/(\mu_t+\lambda_t)v_t$. In other words $k_t = (1+\lambda_t)/(\mu_t+\lambda_t)$. Note that k_t is multiplicatively *inseparable* across λ_t and μ_t, therefore, we need a centralized pacing to update k_t.

The dual variables are updated using feedback loops based on the auction result that have natural self-correcting features to prevent constraint violations (see Algorithm 1). For example, in the case of the budget constraint, the feedback loop in (5) seeks to equate the actual spend of the auction $p_t(b_t)$ with the per-round budget ρ to satisfy the budget constraint (whenever this constraint is binding). Mathematically, the budget and ROS dual variables are updated in (4) and (5) using multiplicative weight updates with the "gradients" set to be the per-period constraint violations of the budget and ROS dual variables. We refer the reader to (Balseiro and Gur, 2019; Feng et al., 2022) for more details. Feng et al. (2022) show that this specific setup obtains near-optimal regret $O(\sqrt{T})$, where regret is the difference between the offline optimal total value and the bidding policy's total value.

Algorithm 1 Dual-Optimal Pacing

Initialize: Initial dual variables $\lambda_1 = 1$, $\mu_1 = 0$, total initial budget $B_1 := \rho T$, gradient descent step-sizes α and η;
for $t = 1, 2, \cdots, T$ **do**

Observe the value v_t, and set the bid

$$b_t = \min\left\{\frac{1+\lambda_t}{\mu_t+\lambda_t}\cdot v_t, B_t\right\}.$$

Update the dual variable of the ROS constraint

$$\lambda_{t+1} := \lambda_t\cdot\exp\Big(-\alpha\cdot(v_t\cdot x_t(b_t) - p_t(b_t))\Big). \tag{4}$$

Update the dual variable of the budget constraint as

$$\mu_{t+1} := \mu_t\cdot\exp\Big(-\eta\cdot(\rho - p_t(b_t))\Big). \tag{5}$$

Update the leftover budget $B_{t+1} = B_t - p_t(b_t)$;

Sequential Pacing. If one were to consider the problem (1) with just the budget constraint, the bidding policy (from Lagrangian duality with the ROS dual variable $\lambda_t = 0$) would be to bid $b_t = v_t/\mu_t$, with the dual variable μ_t alone getting updated as in Algorithm 1. Similarly, if one were to consider the problem (1) with just the ROS constraint, the bidding policy (from Lagrangian duality with budget dual variable $\mu_t = 0$) would be to bid $b_t = v_t \cdot \frac{1+\lambda_t}{\lambda_t}$, with the dual variable λ_t alone updated as in Algorithm 1. Given the historical context mentioned earlier, budget pacing systems have been around for longer than ROS pacing optimization. Therefore, it is not unexpected to have separate servers handling the feedback loops of the budget and ROS constraints and the final bid constructed in a sequential manner, namely,

$$b_t^{\text{seq}} = \min\left\{\frac{1+\lambda_t}{\lambda_t}\cdot\frac{1}{\mu_t}\cdot v_t, B_t\right\}. \tag{6}$$

[2]An auction is truthful if a (profit-maximizing) bidder's utility is maximized when reporting the true value, i.e., $v \in \operatorname{argmax}_b\{v\cdot x_t(b) - p_t(b)\}$. Note the definition of truthful auction is for the quasi-linear utility even though the actual bidder utility considered in the auto-bidding setting is the overall value instead.

[3]If the bidder had access to $x_t(\cdot)$ and $p_t(\cdot)$ before placing the bid at time t, the optimizer is $g_t(v) \in \operatorname{argmax}_b\{v\cdot x_t(b) - p_t(b)\}$.

In other words, the ROS constraint pacing service determines an intermediary bid $\widehat{b}_t = (1+\lambda_t)/\lambda_t \cdot v_t$ which is fed to the budget service and, in turn, the budget pacing service operates on the scaled bid $\widehat{b}_t$ to get the final bid of $\widehat{b}_t/\mu_t$ (and also capped by the remaining budget B_t). While not optimal, this implementation has the benefit of being decentralized, i.e., it could operate separate servers for budget pacing and ROS pacing, that (a) only communicate the temporary bid $\widehat{b}_t$ and (b) could update their respective variables at different frequencies.

Min Pacing. If the transition from sequential to dual-optimal pacing proves prohibitively expensive in the short term for organizational or engineering reasons, we propose and study another decentralized optimization, that we call the *min pacing* service. Rather than applying the bid-lowering operations of the two pacing systems sequentially, we take the minimum of the bids generated by the two systems:

$$b_t^{\min} = \min\left\{\frac{1+\lambda_t}{\lambda_t}\cdot v_t, \frac{1}{\mu_t}\cdot v_t, B_t\right\} . \tag{7}$$

The corresponding dual variables can follow the same update rules in Algorithm 1 (i.e., (4) and (5)). The min pacing service operates in parallel instead of sequentially and also requires minimum coordination between budgeting and ROS pacing. We will show that even though $b_t^{\min}$ is in general different from $b_t^{\text{dual-opt}}$, and thus not the optimizer of the Lagrangian dual (2), bidding $b_t^{\min}$ nonetheless achieves the same asymptotically optimal guarantees on regret and constraint violation as the dual-optimal pacing algorithm.

3. Regret Analysis of the Bidding Algorithms

In this section we analyze the performances of the pacing algorithms introduced in the previous section, and we use the notions of regret and constraint violation. To define the regret, we first define the reward of some pacing algorithm Alg for a sequence of requests $\overrightarrow{\gamma}$ over a time horizon T as

$$\mathsf{Reward}(\mathsf{Alg}, \overrightarrow{\gamma}) := \sum_{t=1}^{T} v_t \cdot x_t(b_t), \tag{8}$$

where b_t's are the algorithm's bids. Note the definition doesn't require Alg to satisfy the budget and ROS constraints. Next, we define the optimal reward $\mathsf{Reward}(\mathsf{OPT}, \overrightarrow{\gamma})$ for a sequence $\overrightarrow{\gamma}$ as the optimal objective of the offline optimization problem (1) given $\overrightarrow{\gamma}$. The regret of Alg is

$$\begin{aligned}\mathsf{Regret}(\mathsf{Alg}, \mathcal{P}^T) := \mathbb{E}_{\overrightarrow{\gamma}\sim\mathcal{P}^T}\,[&\mathsf{Reward}(\mathsf{OPT}, \overrightarrow{\gamma}) \\ &- \mathsf{Reward}(\mathsf{Alg}, \overrightarrow{\gamma})] \,.\end{aligned} \tag{9}$$

We remark that we define Reward for some specific drawn sequence, whereas Regret is defined with respect to a distribution.

Note that Regret itself does not fully measure the performance of Alg since the reward of Alg does not capture the budget and ROS constraints. All the pacing algorithms we discuss will cap the bid by the remaining budget, so the budget constraint is always satisfied. To evaluate our algorithms, we first need the following notion of stopping time.

Definition 3.1. *The stopping time τ of Algorithm 1, with budget B is the first time τ at which $\sum_{t=1}^{\tau} p_t(b_t) + 1 \geq B$.*

Intutively, τ is the first time step when the total payment almost exceeds the total budget. We also explore the *budget endurance* of a bidding algorithm, that is, whether $T-\tau$ is small for any $\overrightarrow{\gamma}$ or, in other words, the budget always runs out close to the end of the horizon.

In addition, we focus on the violation of the ROS constraint, i.e. $\sum_{t=1}^{T} p_t(b_t) - \sum_{t=1}^{T} v_t \cdot x_t(b_t)$. For constraint error we look at ex-post guarantees that hold for any $\overrightarrow{\gamma}$.

In particular, both the joint pacing and min pacing algorithms achieve asymptotically nearly optimal guarantees in terms of the regret and constraint error in the stochastic i.i.d. setting. For simplicity we assume in our analysis that the allocation and payment functions are from truthful auctions. The result for the joint algorithm is already known from previous work in (Balseiro et al., 2022b; Feng et al., 2022).

We start with analyzing the regret of $\mathsf{MinPacing}$ by considering a continuous-time approximation of the algorithm in which multipliers are updated using the expected gradients instead of their noisy stochastic counterparts used in the real algorithm.

Before proceeding with our analysis we provide some useful definitions. We denote by

$$\begin{aligned} g^{\mathrm{BUD}}(k) &= \rho - \mathbb{E}_\gamma[p(k\cdot v)] \,, \\ g^{\mathrm{ROS}}(k) &= \mathbb{E}_\gamma[v\cdot x(k\cdot v) - p(k\cdot v)] \end{aligned}$$

the expected error in the budget and ROS constraints when the multiplier is k. Expectations are taken with respect to the random tuple $\gamma = (v, x, p)$. We plot some examples in Figure 3, which is located in the appendix. We require the following assumptions in our analysis.

Our first assumption is that the functions g^{ROS} and g^{BUD} cross zero once and from above, and that they are Lipschitz continuous.

Assumption 3.2. *We assume that the functions $g^{\mathrm{ROS}}(k)$ and $g^{\mathrm{BUD}}(k)$ are L_g-Lipschitz continuous in k and bounded. Moreover, the following hold:*

(1) The function $g^{\mathrm{BUD}}(k)$ crosses the non-negative k-axis once at $k^{\mathrm{BUD}} > 0$ and from above. That is, for any $0 \leq k < k^{\mathrm{BUD}}$, we have $g^{\mathrm{BUD}}(k) > 0$ and for any $k > k^{\mathrm{BUD}}$, we have $g^{\mathrm{BUD}}(k) < 0$.

(2) The function $g^{\mathrm{ROS}}(k)$ crosses the positive $k-$axis once at $k^{\mathrm{ROS}} > 1$ and from above. That is, for any $0 < k <$

k^{ROS}*, we have* $g^{\mathrm{ROS}}(k) > 0$ *and for any* $k > k^{\mathrm{ROS}}$*, we have* $g^{\mathrm{ROS}}(k) < 0$.

When the auction is truthful, it can be shown that the functions g^{ROS} and g^{BUD} always cross the positive axis from above. Therefore, the set of crossing points is always an interval. Assumption 3.2 rules out the possibility of multiple crossing points and, as we shall discuss later, implies the uniqueness of the optimal bidding strategy. This assumption is related to the so-called "general position" condition, which is pervasive in online allocation problems (see, e.g., Devanur and Hayes 2009; Agrawal et al. 2014). The Lipschitz continuity of the gradients is a common assumption in the analysis of online algorithms (Hazan et al., 2016) and holds when either the interim allocation and payment are smooth, or the distribution of values is absolutely continuous. For example, this assumption might fail to hold in a second-price auction when values and competing bids are discrete (there, g^{ROS} and g^{BUD} are piecewise constant). In this case, it is possible to recover Lipschitz continuity by adding a small amount of random noise to the bids, which mollifies the functions g^{ROS} and g^{BUD}, without significantly impacting the performance of our algorithm. As a result, Assumption 3.2 is not too restrictive.

Under Assumption 3.2, we can upper bound the optimal performance in terms of the value collected by a uniform bidding policy that bids the minimum of the multipliers k^{BUD} and k^{ROS}, and provide a simple characterization of an optimal dual solution. The dual problem becomes $\min_{\mu\geq 0,\lambda\geq 0} D(\mu,\lambda)$ where

$$D(\mu,\lambda) := \underset{k\geq 0}{\text{maximize}}\, \{(1+\lambda)\mathbb{E}_\gamma[v\cdot x(k\cdot v)]\rho\cdot\mu \\ -(\lambda+\mu)\mathbb{E}_\gamma[p(k\cdot v)]\}\ ,$$

is the dual function. A proof of Lemma 3.3 is presented in Appendix C.2.

Lemma 3.3. *Suppose Assumption 3.2 holds. There exists an optimal solution with* $\lambda^* = 0$ *and* $\mu^* = 1/k^{\mathrm{BUD}}$ *if* $k^{\mathrm{BUD}} \leq k^{\mathrm{ROS}}$ *or* $\lambda^* = 1/(k^{\mathrm{ROS}}-1)$ *and* $\mu^* = 0$ *if* $k^{\mathrm{ROS}} \leq k^{\mathrm{BUD}}$*. Moreover, with* $k^* = \min(k^{\mathrm{ROS}}, k^{\mathrm{BUD}})$*,*

$$\mathbb{E}_{\vec{\gamma}\sim\mathcal{P}^T}\left[\mathsf{Reward}(\mathsf{OPT},\vec{\gamma})\right] \leq T\cdot D(\mu^*,\lambda^*) \\ = T\cdot\mathbb{E}_\gamma\left[v\cdot x(k^*\cdot v)\right]$$

Assumption 3.4. *The problem is non-degenerate, namely,* $k^{\mathrm{BUD}} \neq k^{\mathrm{ROS}}$.

The non-degeneracy assumption guarantees that only one of the budget constraint or the ROS constraint can be binding for the uniform bidding policy. This type of assumption is common in the online allocation literature (see, e.g., Jasin and Kumar 2012). In practice, the data comes from a random process, and the budget and targets are given by the advertiser. Notice that the degenerate case stays in a lower dimension manifold, thus it is very likely that the non-degenerate assumption holds. Under the non-degeneracy assumption, the optimal multiplier is either $k^* = k^{\mathrm{BUD}}$ or $k^* = k^{\mathrm{ROS}}$. We remark that this assumption is only imposed to simplify the analysis. In Appendix C.4, we discuss how to provide a similar regret bound of $\sqrt{T}$ for the degenerate case using techniques similar to the ones presented in this paper.

Assumption 3.5. *The gradients of the budget and ROS constraints have second moments bounded by* $\overline{G_2}$.

Assumption 3.5 is common in the analysis of first-order algorithms for online optimization, where second moments are usually required to be bounded (Hazan et al., 2016). When the auction is truthful, a sufficient condition for this assumption to hold is that values have bounded second moments, i.e., $\mathbb{E}_\gamma[v^2] < \infty$.

Assumption 3.6. *There exists some* $\delta > 0$ *and* $\ell > 0$ *such that for all* $k \in [k^*-\delta, k^*+\delta]$ *we have that either* $-g^{\mathrm{BUD}}(k)(k-k^*) \geq \ell\cdot(k-k^*)^2$ *if* $k^* = k^{\mathrm{BUD}}$ *or* $-g^{\mathrm{ROS}}(k)(k-k^*) \geq \ell\cdot(k-k^*)^2$ *if* $k^* = k^{\mathrm{ROS}}$.

Our final assumption requires that the spend and conversion value are locally strongly monotone around the optimal solution. In other words, we require the gradients to be locally linear around point where they cross the positive axis, for example, when the budget constraint is binding, we require that $g^{\mathrm{BUD}}(k) \approx k^* - k$ around k^* (see Fig 3). Similarly to our Lipschtiz condition, we can guarantee this assumption holds by randomly perturbing bids. Assumption 3.6 is also common in the analysis of online algorithms, where it is sometimes assumed that objective functions are strongly convex, which is equivalent to gradients being strongly monotone. See chapter 2.3 of Hazan et al. (2016) for a discussion on handling functions that are not strongly convex problems in online optimization algorithms by reducing them to the strongly convex case.

The next theorem shows that the MinPacing algorithm has an $O\left(T^{1/2}\right)$ regret bound.

Theorem 1. *Suppose Assumption 3.2-3.6 hold. Then, the regret of* MinPacing *can be bounded as:*

$$\mathsf{Regret}(\mathsf{MinPacing},\mathcal{P}^T) = O\left(T^{1/2}\right)\ .$$

One can show that the optimal joint algorithm also has $O\left(T^{1/2}\right)$ regret bound, which showcases that the MinPacing algorithm achieves good practical performance.

Analyzing the min algorithm is challenging as we do not have access to a Lyapunov function as in the joint pacing case. We analyze the algorithm by carefully studying the dynamics of the dual variables under the min algorithm.

Note that in light of Lemma 3.3, if we knew in advance which constraint is binding, then we could attain low regret by bidding using the multiplier associated with the binding constraint ($k = 1/\mu$ for the budget constraint and $k = (1+\lambda)/\lambda$ for the ROS constraint). Our proof technique is to show that with high probability the algorithm detects in $\sqrt{T}$ steps which constraint is binding and then bids according to the optimal bidding multiplier for the binding constraint. A detailed proof of Theorem 1 is presented in Appendix C.3.

Next, we present the constraint violation of the MinPacing algorithm. Recall that the bid is capped by the remaining budget; thus, the budget constraint will always be satisfied. Instead, we show that the budget always runs out $O(\sqrt{T})$ close to the horizon's end. On the other hand, for ROS constraint, we allow small violations throughout the horizon, and we can show that the violation is at most $O(\sqrt{T})$. The proof of Theorem 2 can be found in Appendix C.5.

Theorem 2. *Suppose payments are at most the bid, i.e., $p_t(b) \leq b \cdot x_t(b)$ for all $b \geq 0$. Then,* MinPacing *satisfies the following:*

(ROS constraint.) *The violation of the ROS constraint is at most $O(\sqrt{T}\log T)$, i.e., $\sum_{t=1}^{T} p_t(b_t) - v_t \cdot x_t(b_t) \leq O(\sqrt{T}\log T)$.*

(Budget endurance.) *The budget always runs out close to the end of the horizon, i.e., stopping time τ satisfies $T - \tau \leq O(\sqrt{T})$.*

Finally, we show that the sequential algorithm fails to work—it may have $\Omega(T)$ regret and/or $\Omega(T)$ constraint violation. The proof of Proposition 3.7 is presented in Appendix D.

Proposition 3.7. *For any initial dual variables μ_0, λ_0 and step-sizes η, α, there is an instance on which the algorithm either violates the ROS constraint by at least $\Omega(T)$ or has a regret at least $\Omega(T)$.*

4. Empirical Study

We empirically evaluate the three algorithms discussed in Section 2.1. For confidentiality and advertiser privacy reasons, we use a semi-synthetic dataset based on actual advertising auctions from an online platform, where the advertising campaigns all use a bidding product which is captured by our optimization formulation (1). In particular, an advertiser bids (and pays) for clicks, i.e., submits bids for cost-per-click, and each click comes with an expected (or predicted) number of conversions (conditioned on the click) denoted as $pconv$, e.g. purchase at advertiser's site. The advertiser has an input target cost per conversion (denoted as $tcpa$), which can be considered as how much the advertiser values each conversion, so the advertiser's (expected) value of a click is the product of $tcpa$ and the click's $pconv$. The advertiser's objective is to maximize the total value over won clicks with constraints on total spend being below an input budget and average cost per conversion below $tcpa$. In our formulation (1), this corresponds to: (i) The value v_t is equal to $tcpa \cdot pconv_t$ (note both $tcpa$ and $pconv_t$ are taken to be independent of the bid; while it is obvious for $tcpa$ to be independent of the bid, $pconv$'s independence is supported by empirical studies (Varian, 2009)); (ii) The allocation $x_t(b_t)$ is the number of clicks won by the advertiser at a bid of b_t; (iii) The payment $p_t(b_t)$ is the cost of the clicks won at a bid of b_t. We denote the advertiser's total objective value as $conv\_val$.

Semi-synthetic Dataset. Since we study the stochastic setting where the functions $x_t(\cdot), p_t(\cdot)$ are drawn i.i.d. from some distribution, our dataset consists of a set of generative models. The parameters of the generative model for any given (actual) advertising campaign we study are derived from the performance of that campaign in the (actual) auction. We discuss the generative model in more details in Appendix E.1.

Evaluation Setup. Our dataset includes 10^5 randomly selected campaigns, and for each campaign, we set the budget constraint (i.e. ρT in (1)) using its actual daily budget B. We divide the day into 10-minute periods and use $T = 144$. For each campaign, we simulate an algorithm 10 times to take the average total $spend$ and total $conv\_val$ (i.e., conversion value) as the result of the algorithm on that campaign. We include more details on how we simulate the algorithms as well as visualizations in Appendix E.2. For each algorithm, we evaluate it on all the campaigns to get 10^5 simulated $(spend, conv\_val)$ pairs, and arrange them into buckets based on the relative ROS constraint error $\max(0, spend/conv\_val - 1)$. We look at the cumulative total $conv\_val$ (and number of campaigns) achieved by the algorithm through the ROS violation buckets. That is, for the bucket of at most $z\%$ relative error in the ROS constraint, we get the total value over all campaigns such that the algorithm has a relative ROS violation of at most $z\%$. The cumulative total value (and number of campaigns) over ROS error buckets gives us the picture of how an algorithm performs with respect to both the optimization objective and the constraints.

Benchmark. For each campaign, our benchmark is the fluid relaxation of (1), but restricted to uniform bidding, i.e., $b_t = k \cdot v_t$ for all t.

$$\begin{aligned} \underset{k\geq 0}{\text{maximize}} \quad & \textstyle\sum_{t=1}^{T} \mathrm{E}[v_t \cdot x_t(k \cdot v_t)] \\ \text{subject to} \quad & \textstyle\sum_{t=1}^{T} \mathrm{E}[p_t(k \cdot v_t)] \leq \sum_{t=1}^{T} \mathrm{E}[v_t x_t(k \cdot v_t)], \\ & \textstyle\sum_{t=1}^{T} \mathrm{E}[p_t(k \cdot v_t)] \leq \rho T\,. \end{aligned} \tag{10}$$

We defer details of the benchmark to Appendix E.3.

Table 2. Cumulative fraction of campaigns and total conversion (normalized by total benchmark) over the ROS relative error buckets.

		Relative Constraint Violation											
	Alg	$(\le)0$	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	∞
Frac. of Campaigns	Dual-optimal	0.62	0.71	0.75	0.78	0.80	0.82	0.83	0.84	0.85	0.86	0.87	1.00
	Min.	0.49	0.64	0.72	0.77	0.80	0.81	0.83	0.84	0.85	0.86	0.87	1.00
	Seq.	0.11	0.15	0.18	0.22	0.26	0.29	0.32	0.35	0.38	0.40	0.43	1.00
Cum. Total Value	Dual-optimal	0.62	0.75	0.77	0.81	0.83	0.84	0.84	0.85	0.85	0.85	0.86	0.88
	Min.	0.42	0.73	0.82	0.86	0.88	0.89	0.89	0.90	0.91	0.91	0.91	0.94
	Seq.	0.19	0.23	0.27	0.30	0.33	0.36	0.38	0.40	0.42	0.44	0.46	1.44

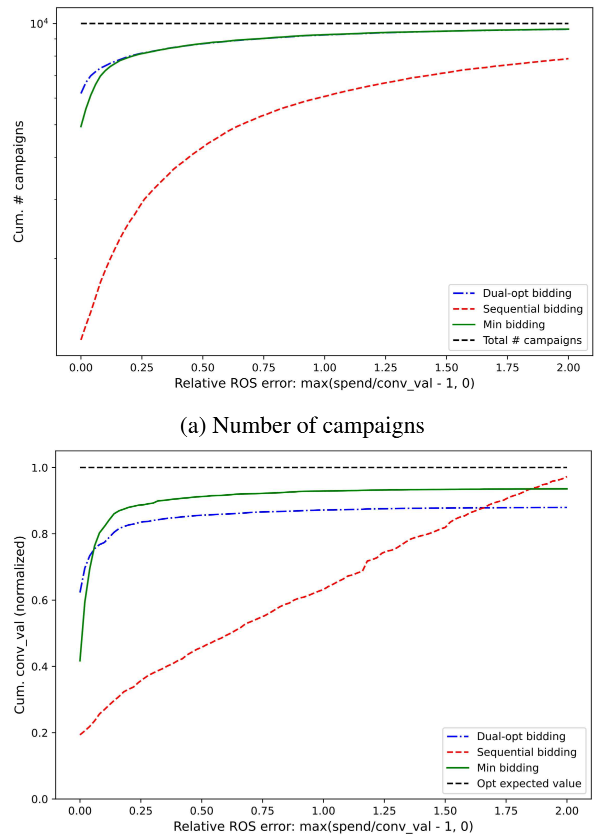
4.1. Results

We show the performance of the three algorithms in Table 2, where each column is associated with a particular error bound, and we show the cumulative fraction of campaigns (top) and cumulative total value of campaigns (bottom) with relative ROS error up to the bound in each column. We normalize the quantities in the table: for value we normalize by our benchmark and for number of campaigns we normalized by the total number 10^5. We look at the results both in terms of how well the algorithms respect the ROS constraint, and also the optimization objective of value maximization. We discuss the stability and convergence of the bidding multipliers generated by the algorithms in Appendix E.4.

ROS constraint. Both the dual-optimal and min pacing algorithms perform well at keeping the relative ROS error reasonably small, e.g., both have a reasonably large 80% of campaigns finish with at most 20% relative ROS error. The sequential pacing algorithm performs poorly in obeying the ROS constraint: only around 11% of campaigns satisfy the ROS constraint, and in Figure $2(a)$ we see a considerable fraction $> 20\%$ of campaigns spend more than twice the conversion value.

Value maximization. Both the dual-optimal and min pacing algorithms also do well at achieving good value. Recall our benchmark on each campaign should be fairly close to the expected optimal value of the fluid relaxation where both the ROS constraint and budget constraint are satisfied on expectation, so it is roughly an upper bound on the expected offline or hindsight optimal, and will be especially meaningful when an algorithm also obey the constraints relatively well. The dual-optimal pacing and min algorithms both achieve very large fraction of the benchmark with fairly small ROS error, e.g., for dual-optimal pacing the campaigns with $\le 15\%$ relative ROS error in total get 81% of the total benchmark conversion values over all campaigns, and for min pacing it is 86% of the total benchmark. The sequential pacing algorithm gets much smaller total value compared to the dual-optimal and min pacing algorithms over campaigns finishing with small ROS error.

Stability and Convergence. We observe that the trajectory of bidding multipliers generated by the dual-optimal and min pacing algorithms converge to the optimal solution of the benchmark (10). On the other hand, the bid multipliers generated by the sequential pacing algorithm can be highly unstable. We include the visualization and discussion of the behavior of the three algorithms on a representative campaign in Appendix E.4 (Figure 8).



(a) Number of campaigns

(b) Total $conv\_val$

Figure 2. Cumulative number of campaigns and total $conv\_val$ for each algorithm over the ROS relative error buckets.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Gagan Aggarwal, Ashwinkumar Badanidiyuru, and Aranyak Mehta. 2019. Autobidding with Constraints. In *Web and Internet Economics - 15th International Conference, WINE 2019, New York, NY, USA, December 10-12, 2019, Proceedings (Lecture Notes in Computer Science, Vol. 11920)*. Springer, 17–30.

Shipra Agrawal and Nikhil R. Devanur. 2015. Fast Algorithms for Online Stochastic Convex Programming. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, Piotr Indyk (Ed.). SIAM, 1405–1424.

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. 2014. A Dynamic Near-Optimal Algorithm for Online Linear Programming. *Oper. Res.* 62, 4 (2014), 876–890.

Zeyuan Allen-Zhu and Lorenzo Orecchia. 2014. Using optimization to break the epsilon barrier: A faster and simpler width-independent algorithm for solving positive linear programs in parallel. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 1439–1456.

Santiago Balseiro, Negin Golrezaei, Mohammad Mahdian, Vahab Mirrokni, and Jon Schneider. 2019. Contextual Bandits with Cross-Learning. In *Advances in Neural Information Processing Systems 32*. 9679–9688.

Santiago Balseiro and Vahab Mirrokni. 2022. Robust Online Allocation with Dual Mirror Descent. https://ai.googleblog.com/2022/09/robust-online-allocation-with-dual.html.

Santiago R. Balseiro, Omar Besbes, and Gabriel Y. Weintraub. 2015. Repeated Auctions with Budgets in Ad Exchanges: Approximations and Design. *Manag. Sci.* 61, 4 (2015), 864–884.

Santiago R. Balseiro, Yuan Deng, Jieming Mao, Vahab S. Mirrokni, and Song Zuo. 2021a. The Landscape of Auto-bidding Auctions: Value versus Utility Maximization. In *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, Péter Biró, Shuchi Chawla, and Federico Echenique (Eds.). ACM, 132–133.

Santiago R. Balseiro, Yuan Deng, Jieming Mao, Vahab S. Mirrokni, and Song Zuo. 2021b. Robust Auction Design in the Auto-bidding World. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 17777–17788.

Santiago R. Balseiro, Yuan Deng, Jieming Mao, Vahab S. Mirrokni, and Song Zuo. 2022a. Optimal Mechanisms for Value Maximizers with Budget Constraints via Target Clipping. In *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, David M. Pennock, Ilya Segal, and Sven Seuken (Eds.). ACM, 475.

Santiago R. Balseiro and Yonatan Gur. 2019. Learning in Repeated Auctions with Budgets: Regret Minimization and Equilibrium. *Manag. Sci.* 65, 9 (2019), 3952–3968.

Santiago R. Balseiro, Anthony Kim, Mohammad Mahdian, and Vahab S. Mirrokni. 2017. Budget Management Strategies in Repeated Auctions. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 15–23.

Santiago R. Balseiro, Haihao Lu, and Vahab Mirrokni. 2022b. The Best of Many Worlds: Dual Mirror Descent for Online Allocation Problems. *Operations Research* (2022). https://doi.org/10.1287/opre.2021.2242

Santiago R Balseiro, Haihao Lu, Vahab Mirrokni, and Balasubramanian Sivan. 2022c. On Dual-Based PI Controllers for Online Allocation Problems. *arXiv preprint arXiv:2202.06152* (2022).

Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian. 2007. Dynamics of bid optimization in online advertisement auctions. In *Proceedings of the 16th international conference on World Wide Web*. 531–540.

Sébastien Bubeck et al. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning* 8, 3-4 (2015), 231–357.

Meta Business Help Center. [n. d.]. Bid Strategies. https://www.facebook.com/business/help/1619591734742116?id=2196356200683573

Xi Chen, Christian Kroer, and Rachitesh Kumar. 2021. The Complexity of Pacing for Second-Price Auctions. In *Proceedings of the 22nd ACM Conference on Economics and Computation* (Budapest, Hungary) *(EC '21)*. Association for Computing Machinery, New York, NY, USA, 318.

Vincent Conitzer, Christian Kroer, Eric Sodomka, and Nicolas E. Stier-Moses. 2022. Multiplicative Pacing Equilibria in Auction Markets. *Oper. Res.* 70, 2 (mar 2022), 963–989.

Yuan Deng, Jieming Mao, Vahab S. Mirrokni, and Song Zuo. 2021. Towards Efficient Auctions in an Auto-bidding World. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 3965–3973.

Nikhil R. Devanur and Thomas P. Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings 10th ACM Conference on Electronic Commerce (EC-2009), Stanford, California, USA, July 6–10, 2009*, John Chuang, Lance Fortnow, and Pearl Pu (Eds.). ACM, 71–78.

Nikhil R. Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A. Wilkens. 2019. Near Optimal Online Algorithms and Fast Approximation Algorithms for Resource Allocation Problems. *J. ACM* 66, 1 (2019), 7:1–7:41.

Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S. Mirrokni, and Clifford Stein. 2010. Online Stochastic Packing Applied to Display Ad Allocation. In *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I (Lecture Notes in Computer Science, Vol. 6346)*, Mark de Berg and Ulrich Meyer (Eds.). Springer, 182–194.

Jon Feldman, S. Muthukrishnan, Martin Pál, and Clifford Stein. 2007. Budget optimization in search-based advertising auctions. In *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), San Diego, California, USA, June 11-15, 2007*, Jeffrey K. MacKie-Mason, David C. Parkes, and Paul Resnick (Eds.). ACM, 40–49.

Zhe Feng, Swati Padmanabhan, and Di Wang. 2022. Online Bidding Algorithms for Return-on-Spend Constrained Advertisers. https://arxiv.org/abs/2208.13713

Zhe Feng, Chara Podimata, and Vasilis Syrgkanis. 2018. Learning to Bid Without Knowing Your Value. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 505–522.

Giannis Fikioris and Éva Tardos. 2022. Liquid Welfare guarantees for No-Regret Learning in Sequential Budgeted Auctions. *CoRR* abs/2210.07502 (2022).

Jason Gaitonde, Yingkai Li, Bar Light, Brendan Lucier, and Aleksandrs Slivkins. 2022. Budget Pacing in Repeated Auctions: Regret and Efficiency without Convergence. https://arxiv.org/abs/2205.08674

Google. Accessed 10/01/2023a. Automated Bidding Strategies. https://support.google.com/google-ads/answer/2979071

Google. Accessed 10/01/2023b. Google Budget Management. https://support.google.com/searchads/answer/7073010#

Anupam Gupta and Marco Molinaro. 2014. How Experts Can Solve LPs Online. In *Algorithms - ESA 2014 - 22th Annual European Symposium, Wroclaw, Poland, September 8-10, 2014. Proceedings (Lecture Notes in Computer Science, Vol. 8737)*, Andreas S. Schulz and Dorothea Wagner (Eds.). Springer, 517–529.

Yanjun Han, Zhengyuan Zhou, Aaron Flores, Erik Ordentlich, and Tsachy Weissman. 2020. Learning to Bid Optimally and Efficiently in Adversarial First-price Auctions. *CoRR* abs/2007.04568 (2020). arXiv:2007.04568

Elad Hazan et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization* 2, 3-4 (2016), 157–325.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. 1–9.

Stefanus Jasin and Sunil Kumar. 2012. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* 37, 2 (2012), 313–345.

Jiashuo Jiang, Xiaocheng Li, and Jiawei Zhang. 2020. Online Stochastic Optimization with Wasserstein Based Non-stationarity. *CoRR* abs/2012.06961 (2020). https://arxiv.org/abs/2012.06961

Yuchin Juan, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. 2016. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM conference on recommender systems*. 43–50.

Thomas Kesselheim, Andreas Tönnis, Klaus Radke, and Berthold Vöcking. 2014. Primal Beats Dual on Online Packing LPs in the Random-Order Model. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (New York, New York) *(STOC '14)*. Association for Computing Machinery, New York, NY, USA, 303–312.

Bhuvesh Kumar, Jamie Morgenstern, and Okke Schrijvers.

2022. Optimal Spend Rate Estimation and Pacing for Ad Campaigns with Budgets. *CoRR* abs/2202.05881 (2022). https://arxiv.org/abs/2202.05881

Harold Joseph Kushner. 1967. *Stochastic stability and control*. Vol. 33. Academic press New York.

Bin Li, Xiao Yang, Daren Sun, Zhi Ji, Zhen Jiang, Cong Han, and Dong Hao. 2020b. Incentive mechanism design for roi-constrained auto-bidding. *arXiv preprint arXiv:2012.02652* (2020).

Xiaocheng Li, Chunlin Sun, and Yinyu Ye. 2020a. Simple and fast algorithm for binary integer and online linear programming. *Advances in Neural Information Processing Systems* 33 (2020), 9412–9421.

Christopher Liaw, Aranyak Mehta, and Andrés Perlroth. 2022. Efficiency of non-truthful auctions under auto-bidding. *CoRR* abs/2207.03630 (2022).

Quan Lu, Shengjun Pan, Liang Wang, Junwei Pan, Fengdan Wan, and Hongxia Yang. 2017. A practical framework of conversion rate prediction for online display advertising. In *Proceedings of the ADKDD'17*. 1–9.

Brendan Lucier, Sarath Pattathil, Aleksandrs Slivkins, and Mengxiao Zhang. 2023. Autobidders with Budget and ROI Constraints: Efficiency, Regret, and Pacing Dynamics. *CoRR* abs/2301.13306 (2023). https://doi.org/10.48550/arXiv.2301.13306 arXiv:2301.13306

H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1222–1230.

Aranyak Mehta. 2022. Auction Design in an Auto-bidding Setting: Randomization Improves Efficiency Beyond VCG. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 173–181.

Meta. Accessed 10/01/2023. About pacing: Meta Business Help Center. https://www.facebook.com/business/help/1754368491258883?id=561906377587030

Roger B. Myerson. 1981. Optimal Auction Design. *Mathematics of Operations Research* 6, 1 (1981), 58–73.

Yury Smirnov, Quan Lu, and Kuang-chih Lee. 2016. Online ad campaign tuning with pid controllers. US Patent App. 14/518,601.

Michael Tashman, Jiayi Xie, John Hoffman, Lee Winikor, and Rouzbeh Gerami. 2020. Dynamic bidding strategies with multivariate feedback control for multiple goals in display advertising. *arXiv preprint arXiv:2007.00426* (2020).

Twitter. Accessed 10/01/2023. How we built Twitter's highly reliable ads pacing service. https://blog.twitter.com/engineering/en_us/topics/infrastructure/2021/how-we-built-twitter-s-highly-reliable-ads-pacing-service

Hal Varian. 2009. Conversion Rates Don't Vary Much with Ad Position. https://adwords.googleblog.com/2009/08/conversion-rates-dont-vary-much-with-ad.html.

Jonathan Weed, Vianney Perchet, and Philippe Rigollet. 2016. Online learning in repeated auctions. In *Conference on Learning Theory*. PMLR, 1562–1583.

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.

Zikun Ye, Dennis Zhang, Heng Zhang, Renyu Philip Zhang, Xin Chen, and Zhiwei Xu. 2020. Cold Start to Improve Market Thickness on Online Advertising Platforms: Data-Driven Algorithms and Field Experiments. *Available at SSRN 3702786* (2020).

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.

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.

Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. 2008. Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems. In *Proceedings of the 17th International Conference on World Wide Web* (Beijing, China) *(WWW '08)*. Association for Computing Machinery, New York, NY, USA, 1243–1244.

A. Related Work

Traditional auction theory in microeconomics studies maximizing objectives such as welfare, revenue and gains from trade in the presence of buyer(s) with quasilinear utility, namely, a utility of $v - p$ where v is the value derived and p be the payment. In this work, we adopt a different behavioral model, namely, one where advertisers maximize their value, subject to constraints on the return-on-spend (ROS) and total budget. As mentioned earlier, the significant rise in the adoption of autobidding algorithms in the past few years (Center, [n. d.]; Google, 2023a) motivates the study of this model.

Optimal bidding algorithm for a single value-maximizing bidder with budget and/or ROS constraints. Aggarwal et al. (2019) initiated the study of value-maximizing bidders (value maximizers for short) subject to quite general constraints on value and cost. In particular, their model includes budget and ROS constraints. They show how the uniform bidding strategy is optimal if and only if the underlying auction is truthful (where truthfulness is defined from the point-of-view of a quasilinear bidder). Closest to our work is (Feng et al., 2022) who study the advertiser's value maximization problem in the presence of both budget and ROS constraints in an online repeated auction setting. They show that a specific instantiation of what we call the joint pacing algorithm in this work achieves a $O(\sqrt{T} \log T)$ regret while respecting both the budget and RoS constraints in the stochastic i.i.d. setting. Their algorithm computes the bid as a function of the two Lagrange multipliers exactly as in Equation (3).

Welfare in equilibrium among value maximizers. While the description so far, and also our work, focuses on a single bidder's optimal bidding problem, the equilibrium under the presence of multiple value maximizing bidders has also been a very active area recently. Aggarwal et al. (2019) show how the VCG mechanism, which is welfare maximizing with quasilinear utility maximizers, can achieve, in the worst case, only a fraction $\frac{1}{2}$ of the optimal social welfare. Recent work by Mehta (2022) shows how randomization can improve the efficiency beyond the $\frac{1}{2}$ guaranteed by VCG, by establishing a POA of 1.89 for 2 bidders and how the POA is unimprovable beyond 2 even with randomized mechanisms when $n \to \infty$. (Liaw et al., 2022) study whether non-truthfulness can improve the POA beyond 2 and show that this is not possible with a deterministic mechanism. But with the combined power of randomization and non-truthful mechanisms, they show how a randomized first-price auction can improve the POA to 1.8 for two bidders, but again show it is unimprovable beyond 2 when the number of bidders is large. Departing from the no information case studied by the above referenced papers, recent works by Balseiro et al. (2021b); Deng et al. (2021) show how to improve the efficiency under equilibrium beyond $\frac{1}{2}$ by adding boosts and reserves respectively, based on additional information from machine learned advice.

Revenue-optimal auction for value maximizers with budget and/or ROS constraints. Much like the design of optimal auctions for utility-maximizing bidders (Myerson, 1981), a recent line of work has focused on the design of revenue optimal mechanisms for value maximizers. Balseiro et al. (2021a); Li et al. (2020b) initiate this line of work, studying the revenue optimal mechanism in the presence of RoS constraints, but no budget constraints, under various information structures regarding whether or not the value is private, whether or not the advertiser specified target is private. (Balseiro et al., 2022a) extend this work to include budget constraints for advertisers, and consider the information structure where value is public, so are advertiser budgets, but advertiser specified target is private.

Optimal bidding algorithm for a single utility maximizing bidder with & without budget constraint. While works dealing with budget and ROS constraints in the presence of value maximizers have already been discussed, there has been a long line of work on doing the same for utility maximizers, but usually with just budget constraints. When values and competing bids are drawn from i.i.d. distributions, Balseiro and Gur (2019) show that the dual subgradient descent algorithm gives the optimal $O(\sqrt{T})$ regret, and in the adversarial setting they show that it obtains the optimal asymptotic competitive ratio, namely, B/T divided by the maximum value. Zhou et al. (2008) also study pacing in the adversarial setting and give an optimal competitive ratio, but one that is differently parameterized compared to (Balseiro and Gur, 2019). Kumar et al. (2022) study an episodic setting and show how to compute per-period target expenditures based on estimating the probability density based on samples, and ultimately pace based on these target expenditures. On similar lines Jiang et al. (2020) also show how to obtain the optimal $\sqrt{T}$ regret in a non-stationary setting by first learning the probability distributions and then computing target expenditures based on those, using $T \log T$ samples per distribution. Our paper is also loosely related with the rich literature about *Learning to bid in repeated auctions* (Borgs et al., 2007; Weed et al., 2016; Feng et al., 2018; Balseiro et al., 2019; Han et al., 2020), in which the existing papers usually abstract this problem as contextual bandits and do not incorporate budget or ROS constraints into them.

Equilibrium among budget-pacing strategies of utility maximizers. There is a line of work studying equilibrium outcomes of budget pacing agents interacting with each other. We refer the reader to (Gaitonde et al., 2022; Fikioris and Tardos, 2022; Chen et al., 2021; Conitzer et al., 2022; Balseiro et al., 2015) and the references therein for more on this topic. Interestingly, these papers show that uniform bidding is also optimal in the presence of budget constraints. Also, Balseiro et al. (2017) perform a comprehensive study of different common budget-pacing strategies and compare the system equilibrium in terms of their welfare, platform revenue, and advertiser utility.

Online resource allocation problems. The budget pacing problem discussed in the preceding paragraphs is known to be a special case of online resource allocation problems, which have a long line of work. Most of the literature on this topic has focused on the i.i.d. input model or the slightly more general random permutation model. Devanur and Hayes (2009) introduce a training-based algorithm that learns the optimal dual variables from a batch of initial requests and then uses those to assign the rest of the requests. They show how to obtain a $O(T^{2/3})$ regret for the budgeted allocation problem (also known as the adwords problem) in the random permutation model. Feldman et al. (2010) obtain a $O(T^{2/3})$ regret for more general linear packing problems in the random permutation model. Agrawal et al. (2014) obtain an improved $O(\sqrt{T})$ regret by repeatedly solving for the optimal dual variables at geometrically increasing time lengths. The algorithm of Kesselheim et al. (2014) further solves a linear program at every step and apart from $O(\sqrt{T})$, also obtain the optimal dependence on the number of resources. Devanur et al. (2019) consider more general online packing and covering LPs, but in the i.i.d. model and obtain a $O(\sqrt{T})$ regret with the optimal dependence on the number of resources. Their algorithm does not need to solve auxiliary linear programs if given an estimate of OPT. (Gupta and Molinaro, 2014; Agrawal and Devanur, 2015; Balseiro et al., 2022b) make the formal connection between dual descent algorithms and online resource allocation, and show how one can use dual descent algorithms as a black box to obtain a $O(\sqrt{T})$ regret. In particular, (Balseiro et al., 2022b; Li et al., 2020a) present simple algorithms that do not require solving auxiliary optimization problems.

B. Practicality considerations of pacing services

As we discussed in the introduction, while the fully coupled joint-pacing and minimally coupled min-pacing enjoy similar performance guarantees, a significant advantage of the min-pacing algorithm is its ease of implementation in practice, which is also the main motivation for us to analyze it theoretically.

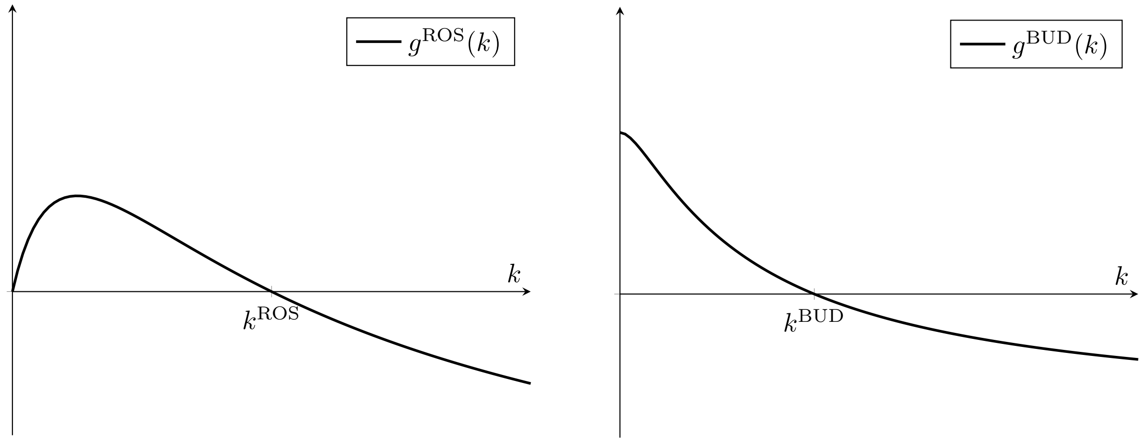
If the advertiser uses an auto-bidding service from a third party and the budget pacing of the platform, anything other than the min-pacing (i.e. the platform takes the min of the in-house budget-bid and the third-party ROS-bid) would likely require the third party to share its inner bidding logic with the platform, which is also fairly infeasible.

Even if the advertiser lets the platform handle both budget and ROS constraints, it can be fairly infeasible to merge these services beyond simple things like the min-pacing that takes a min of two bids before auction. One may imagine a scenario that instead of sending the two bids, the servers can send the two dual variables λ_t, μ_t and compute the final bid according to the dual-optimal joint-pacing formula before auction. This is plausible if all the products admit such a simple bidding formula. However, even though our optimization formulation captures many of the most popular auto-bidding products on the market, the auto-bidding teams still manage many (smaller) products with constraints or implicit business logic not under our model, and the bidding formulas can be more complicated with many other dual variables depending on the product. This makes it much more expensive to pass through all dual variables (or similarly to pass through the final bid for some subset of products but the dual variables for other products) and apply complicated product-specific bidding logic before auction, and thus it's much more expensive in general for the platform to implement the dual-optimal joint-bidding compared to min-pacing.

C. Proofs of Main Results

C.1. Proof Sketch of Main Result

To do so, we consider a continuous time approximation of the multipliers $(\overline{\lambda}(s)), \overline{\mu}(s))$ in which we update them continuously according to the expected gradients, and dynamics are governed by an ODE. The ODE traces the "expected" path of the multipliers when the step-size is small. Here, we assume that step-sizes are $\alpha > 0$ for both constraints. The



(a) Expected gradient of the ROS constraint $g^{\mathrm{ROS}}(k) = \mathbb{E}_v[v \cdot x(k \cdot v) - p(k \cdot v)]$.

(b) Expected gradient of the budget constraint $g^{\mathrm{BUD}}(k) = \rho - \mathbb{E}_v[p(k \cdot v)]$.

Figure 3. Expected gradients for an example in which values and competing bids are independent and exponentially distributed with means 1/2 and 1, respectively. Also, $\rho = 9/16$. Both curves cross the positive k-axis once and from above (Assumption 3.2). Also, strong monotonicity holds for this example (Assumption 3.6).

ODEs are obtained by considering the continuous approximation of multiplicative weight updates:

$$\frac{d}{ds}\log(\overline{\lambda}(s)) = -g^{\mathrm{ROS}}\left(k^{\min}(\overline{\lambda}(s), \overline{\mu}(s)\right), \quad (11)$$
$$\frac{d}{ds}\log(\overline{\mu}(s)) = -g^{\mathrm{BUD}}\left(k^{\min}(\overline{\lambda}(s), \overline{\mu}(s)\right),$$

where

$$k^{\min}(\lambda, \mu) = \min\left(\frac{1+\lambda}{\lambda}, \frac{1}{\mu}\right).$$

The time in the ODE, which is denoted by $s \geq 0$ can be mapped to a step t in the discrete-time stochastic system by setting $s = \alpha t$. In other words, the time in the ODE corresponds to the total distance traveled according to the step-size. We assume throughout that the ROS constraint binds at optimality. A similar analysis holds for the budget constraint. Our proof strategy is the following.

(1) *Binding Constraint Identification.* Setting the step-size to be $\alpha \approx T^{-1/2}$, we show it takes order $\sqrt{T}$ steps to be get to an orbit of size ϵ of an dual optimal solution $(\lambda^*, 0)$. The orbit is chosen so that the bidding formula in this region is $k = (1+\lambda)/\lambda$. We do so by first arguing that the ODE gets in a constant amount of time to the orbit of the optimal and then arguing that the actual algorithm remains close to the expected path traced by the ODE with probability $T^{-1/2}$ using a discrete version of Gronwall's Lemma to bound the absolute deviations and then invoking a concentration argument to bound the maximum deviation in a stochastic sense.

(2) *Orbital Stability.* Once the algorithm reaches an orbit of an optimal solution, we show that it never leaves the orbit with probability $T^{-1/2}$. We prove this result by constructing a local stochastic Lyaponuv function using the KL divergence and then invoking a classical result from stochastic stability.

(3) *Regret Analysis.* We conclude by showing that the regret accumulated once the algorithm is in the orbit of the optimal solution is $\sqrt{T}$. For this step, we first lower bound the conversion value collected by the algorithm in terms of the dual function $D(\lambda, \mu)$ and a complementary slackness term. Using weak duality, we can relate the first term to the optimal performance. The complementary slackness term is controlled using standard regret bounds for multiplicative weight updates.

C.2. Proof of Lemma 3.3

We first prove the upper bound on OPT and then the characterization of the dual optimal solution.

Part 1 (weak duality). The upper bound follows from weak duality. Let $\lambda \geq 0$ and $\mu \geq 0$ be the Lagrange multipliers for the ROS and budget constraints, respectively. Moving the corresponding constraints in (1) to the objective, we obtain the upper bound

$$\begin{aligned}
\mathbb{E}_{\vec{\gamma}\sim\mathcal{P}^T}\left[\mathsf{Reward}(\mathsf{OPT},\vec{\gamma})\right] &\leq \mathbb{E}_{\vec{\gamma}\sim\mathcal{P}^T}\left[\max_{b_t:t=1,\cdots,T}\left\{T\rho\mu + \sum_{t=1}^{T}\Big((1+\lambda)\cdot v_t\cdot x_t(b_t) - (\lambda+\mu)\cdot p_t(b_t)\Big)\right\}\right] \\
&= T\rho\mu + T\mathbb{E}_\gamma\left[\max_b\left\{(1+\lambda)\cdot v\cdot x(b) - (\lambda+\mu)\cdot p(b)\right\}\right] \\
&= T\left(\rho\mu + \max_{k'\geq 0}\mathbb{E}_\gamma\left[(1+\lambda)\cdot v\cdot x(k'\cdot b) - (\lambda+\mu)\cdot p(k'\cdot v)\right]\right) \\
&= T\cdot D(\mu,\lambda)\,,
\end{aligned}$$

where the first equation follows from using that the tuples $\gamma_t = (v_t, x_t, p_t)$ are drawn i.i.d. and that the problem is separable, and the last equation because bidding a uniform multiplier of the value is optimal by (3).

Part 2 (characterization of dual optimal solutions). The characterization of the dual solution follows because there exists an optimal solution in which either the dual variable of the budget constraint or the RoS constraint is zero. To see this, fix some $k > 0$ and let $C_k = \{(\mu,\lambda)\in\mathbb{R}^2_+ : k = (1+\lambda)/(\lambda+\mu)\}$ be all dual feasible solutions with a multiplier of k. When the auction is truthful, we can write the dual function for $(\mu,\lambda)\in C_k$ as follows

$$\begin{aligned}
D(\mu,\lambda) &= \max_{k'\geq 0}\left\{(1+\lambda)\mathbb{E}_\gamma[v\cdot x(k'\cdot v)] + \rho\cdot\mu - (\lambda+\mu)\mathbb{E}_\gamma[p(k'\cdot v)]\right\} \\
&= \rho\cdot\mu + (\lambda+\mu)\cdot\max_{k'\geq 0}\left\{\mathbb{E}_\gamma\left[\frac{1+\lambda}{\lambda+\mu}\cdot v\cdot x(k'\cdot v) - p(k'\cdot v)\right]\right\} \\
&= \rho\cdot\mu + (\lambda+\mu)\cdot\mathbb{E}_v\left[k\cdot v\cdot x(k\cdot v) - p(k\cdot v)\right]\,,
\end{aligned}$$

where the second equation follows from extracting constant terms and the last because the bidding $k\cdot v$ is a best-response for a bidder in a truthful auction in which their value is $k\cdot v$. The set C_k is affine. Moreover, once we fix the value of k, the dual objective $D(\mu,\lambda)$ is linear in the dual variables and, by individual rationality, the coefficients of the dual variables are non-negative. Therefore, the dual function attains its maximum at an extreme point of $D(\mu,\lambda)$. Note that the extreme points of C_k have either $\mu = 0$ or $\lambda = 0$ since C_k is defined by the constraints $\mu \geq 0$, $\lambda \geq 0$, and the hyperplane $k\cdot\mu + (k-1)\cdot\lambda = 1$. We now consider each case at a time.

Suppose at an optimal dual solution we have $\mu^* = 0$. By the Envelope Theorem we have that $\partial D(\mu,\lambda)/\partial\lambda = \mathbb{E}_v\left[v\cdot x(k\cdot v) - p(k\cdot v)\right] = g^{\mathrm{ROS}}(k)$. Therefore, Assumption 3.2 implies that $\mathrm{argmin}_{\lambda\geq 0} D(\mu,\lambda) = 1/(k^{\mathrm{ROS}} - 1)$ because the first-order conditions are such that $g^{\mathrm{ROS}}(k^{\mathrm{ROS}}) = 0$ and $k^{\mathrm{ROS}} = (1+\lambda^*)/\lambda^*$ since $\mu^* = 0$. Thus, we have that the optimal dual objective value when $\mu^* = 0$ is

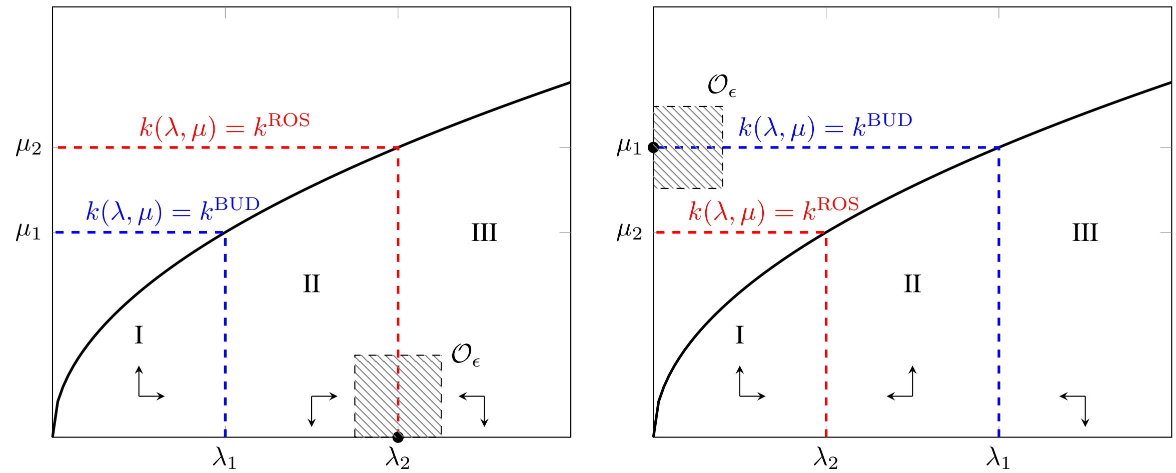
$$\begin{aligned}
\min_{\mu\geq 0,\lambda\geq 0} D(\mu,\lambda) &= D(0,\lambda^*) = (1+\lambda^*)\cdot\mathbb{E}_\gamma[v\cdot x(k^{\mathrm{ROS}}\cdot v)] - \lambda^*\cdot\mathbb{E}_\gamma[p(k^{\mathrm{ROS}}\cdot v)] \\
&= \mathbb{E}_\gamma[v\cdot x(k^{\mathrm{ROS}}\cdot v)] - \lambda^*\cdot g^{\mathrm{ROS}}(k^{\mathrm{ROS}}) = \mathbb{E}_\gamma[v\cdot x(k^{\mathrm{ROS}}\cdot v)]\,,
\end{aligned}$$

where the third equation follows from the definition of g^{ROS} and the last because $g^{\mathrm{ROS}}(k^{\mathrm{ROS}}) = 0$ by Assumption 3.2. A similar argument for the case when $\lambda^* = 0$ implies that the optimal solution in that case is $\mu^* = 1/k^{\mathrm{ROS}}$ and $D(\mu^*, 0) = \mathbb{E}_\gamma[v\cdot x(k^{\mathrm{BUD}}\cdot v)]$.

Combining both cases and using that there always exists an optimal solution with either $\mu^* = 0$ or $\lambda^* = 0$ we conclude that

$$\begin{aligned}
\min_{\mu\geq 0,\lambda\geq 0} D(\mu,\lambda) &= \min\left(\min_{\lambda\geq 0} D(0,\lambda), \min_{\mu\geq 0} D(\mu,0)\right) \\
&= \min\left(\mathbb{E}_\gamma[v\cdot x(k^{\mathrm{ROS}}\cdot v)], \mathbb{E}_\gamma[v\cdot x(k^{\mathrm{BUD}}\cdot v)]\right) = \mathbb{E}_\gamma[v\cdot x(k^*\cdot v)]\,,
\end{aligned}$$

because the allocation is non-decreasing and letting $k^* = \min(k^{\mathrm{ROS}}, k^{\mathrm{BUD}})$. The result follows.



(a) Case 1: ROS constraint is binding, i.e., $k^{\mathrm{ROS}} < k^{\mathrm{BUD}}$. An optimal dual solution is $\lambda^* = 1/(k^{\mathrm{ROS}} - 1)$ and $\mu^* = 0$.

(b) Case 2: budget constraint is binding, i.e., $k^{\mathrm{BUD}} < k^{\mathrm{ROS}}$. An optimal dual solution is $\lambda^* = 0$ and $\mu^* = 1/k^{\mathrm{BUD}}$.

Figure 4. Illustration of the two cases for the MIN dynamics in the non-degenerate case ($k^{\mathrm{BUD}} \neq k^{\mathrm{ROS}}$). The black dot indicates an optimal solution and the hatched rectangle is an orbit $\mathcal{O}_\epsilon$ of size ϵ around the optimal solution. The solid black curve gives the points for which the $(1+\lambda)/\lambda = 1/\mu$, i.e., the multipliers of both constraints are equal. Above the curve, the algorithm bids $1/\mu$ according to the budget constraint, and below it bids $(1+\lambda)/\lambda$ according to the ROS constraint. The arrows indicate the drift of the stochastic process in each region. The red (blue, resp.) dashed curve gives the set of dual variables for which $k^{\min}(\lambda, \mu) = k^{\mathrm{ROS}}$ ($= k^{\mathrm{BUD}}$, resp.).

C.3. Proof of Theorem 1: Analysis of MinPacing Algorithm

We choose the orbit to be a ball of size $\epsilon > 0$ around the optimal solution

$$\mathcal{O}_\epsilon = \left\{ (\lambda, \mu) \in \mathbb{R}_+^2 : \max\left(|\lambda - \lambda^*|, |\mu - \mu^*|\right) < \epsilon \right\} .$$

The value of $\epsilon > 0$ is chosen so that

1. Assumption 3.6 is satisfied for all $(\lambda, \mu) \in \mathcal{O}_\epsilon$,
2. The algorithm bids according to the binding constraint for all $(\lambda, \mu) \in \mathcal{O}_\epsilon$, i.e., $k^{\min}(\lambda, \mu) = (1+\lambda)/\lambda$,
3. The gradient of the budget constraint satisfies $g^{\mathrm{BUD}}(k^{\min}(\lambda, \mu)) > 0$ for all $(\lambda, \mu) \in \mathcal{O}_\epsilon$,
4. The gradients satisfy $g^{\mathrm{ROS}}(k^{\min}(\lambda, \mu)) < 0$ and $g^{\mathrm{BUD}}(k^{\min}(\lambda, \mu)) < 0$ if either $\lambda < \epsilon$ or $\mu < \epsilon$.

The second condition can be satisfied by Lemma 3.3 because there exists an optimal dual optimal solution with $\mu^* = 0$ and $\lambda^* = 1/(k^{\mathrm{ROS}} - 1) > 0$, and the algorithm bids according to the ROS multiplier when $(1+\lambda)/\lambda < 1/\mu$. The third condition can be satisfied because the single-crossing property (Assumption 3.2) implies that $g^{\mathrm{BUD}}(k) > 0$ for $k < k^{\mathrm{BUD}}$ and non-degeneracy (Assumption 3.4) implies that for $k^{\mathrm{ROS}} < k^{\mathrm{BUD}}$. The fourth condition holds by the single-crossing property because the gradients are negative for large enough multipliers k.

C.3.1. Step 1: Binding Constraint Identification

For the first step, we show that if dual variables are positive, it takes the ODE a constant amount of time to get to the interior of the orbit.

Lemma C.1. *For any initial dual solution $(\lambda(0), \mu(0)) \notin [0, \epsilon)^2$, there exits a finite time $\sigma > 0$ such that the solution of* (11) *satisfies $(\lambda(\sigma), \mu(\sigma)) \in \mathcal{O}_{\epsilon/2}$ and for all $s \in [0, \sigma]$ we have $(\lambda(s), \mu(s)) \notin [0, \epsilon)^2$.*

Proof. It follows from Assumption 3.2 and 3.4 that $g^{\mathrm{ROS}}(k) > 0$ for $k < k^{\mathrm{ROS}}$ and $g^{\mathrm{ROS}}(k) < 0$ for $k > k^{\mathrm{ROS}}$. Furthermore, denote $\mu_1 = 1/k^{\mathrm{BUD}}$, $\mu_2 = 1/k^{\mathrm{ROS}}$, $\lambda_1 = 1/(k^{\mathrm{BUD}} - 1)$, $\lambda_2 = 1/(k^{\mathrm{ROS}} - 1)$. We consider two cases depending on whether k^{BUD} or k^{ROS} is smaller.

Case 1: $k^{\mathrm{BUD}} > k^{\mathrm{ROS}}$**.** In this case, we have $k^* = k^{\mathrm{ROS}}$, and the unique stationary point is given by $\mu^* = 0$ and $\lambda^* = \lambda_2 = 1/(k^{\mathrm{ROS}} - 1)$.

The whole space can be split into three regions (see Figure 4a):

Region I: $k > k^{\mathrm{BUD}}$**.** This region corresponds to $\{(\mu, \lambda) : \mu < \mu_1, \lambda < \lambda_1\}$. In this region, we have $g^{\mathrm{ROS}}(k) < 0$ and $g^{\mathrm{BUD}}(k) < 0$, thus $\dot{\mu} > 0$ and $\dot{\lambda} > 0$.

Region II: $k^{\mathrm{ROS}} < k < k^{\mathrm{BUD}}$**.** This region corresponds to $\{(\mu, \lambda) : \mu < \mu_2, \lambda < \lambda_2\}$ subtracting region I. In this region, we have $g^{\mathrm{BUD}}(k) > 0$ and $g^{\mathrm{ROS}}(k) < 0$, thus $\dot{\mu} < 0$ and $\dot{\lambda} > 0$.

Region III: $k > k^{\mathrm{ROS}}$**.** This region corresponds to the complementary set of region I and II. In this region, we have $g^{\mathrm{BUD}}(k) > 0$ and $g^{\mathrm{ROS}}(k) > 0$, thus $\dot{\mu} < 0$ and $\dot{\lambda} < 0$.

Now, we are ready to show the result. Before proceeding, note that by definition ϵ, we have that $\dot{\mu} > 0$ and $\dot{\lambda} > 0$ if $\mu < \epsilon$ and $\lambda < 0$. Therefore, the ODE can never get closer to a distance ϵ from the origin.

First, we claim that for any initial solution $\mu(0), \lambda(0)$, there exists s_1 such that it holds for all $s > s_1$ that $\mu(s) \leq \widehat{\mu} := \frac{1}{2}(\mu_1 + \mu_2)$. This is because once $\mu(s) \leq \widehat{\mu}$, $\mu(s)$ would never go above $\widehat{\mu}$ due to the dynamics in regions II and III. So we just need to consider the first time $\mu(s) \leq \widehat{\mu}$. Notice that for all (μ, λ) such that $\mu > \widehat{\mu}$, there exists δ_1 such that we have $\dot{\mu} < \delta_1 < 0$. Thus, we just need to choose $s_1 = \frac{1}{|\delta_1|}((\mu(0) - \mu_1)^+)$.

Second, we claim there exists $s_2 > s_1$ such that for $s > s_2$, we have $\mu(s) \leq \widehat{\mu}$ and $\lambda(s) \geq \widehat{\lambda} := \frac{1}{2}(\lambda_1 + \lambda_2)$. This is because after s_1, $\mu(s) \leq \widehat{\mu}$. Thus, once $\lambda(s) \geq \widehat{\lambda}$, $\lambda(s)$ would never go below $\widehat{\lambda}$ due to the dynamics in the regions I and II. So we just need to consider the first time $\lambda(s) \geq \widehat{\lambda}$. Notice that for all (μ, λ) such that $\mu \leq \mu_1, \lambda \leq \widehat{\lambda}$, there exists δ_2 such that we have $\dot{\lambda} \geq \delta_2 > 0$. Thus, we just need to choose $s_2 = s_1 + \frac{1}{\delta_2}((\widehat{\lambda} - \lambda(s_1))^+)$.

Third, we claim there exists $s_3 > s_2$ such that for $s > s_3$, we have $\mu(s) \leq \epsilon/2$ and $\lambda(s) \geq \widehat{\lambda}$. This is because after s_2, $\mu(s) \leq \widehat{\mu}, \lambda(s) \geq \widehat{\lambda}$. In this region, there exists $\delta_3 < 0$ such that $\dot{\mu} \leq \delta_3 \mu < 0$ and we just need to choose $s_3 = s_2 + \log(\epsilon/(2\mu))/|\delta_3|$.

Fourth, we claim there exists $s_4 > s_3$ such that for $s > s_4$, we have that $\mu(s) \leq \epsilon/2$ and $|\lambda(s) - \lambda_2| \leq \epsilon/2$. This is because after s_3 we have that $\dot{\mu} \leq 0$ and hence $\mu(s) \leq \epsilon/2$ for all $s > s_3$. The single-crossing property implies that $\dot{\lambda} = 0$ only at λ_2, so we should reach $|\lambda(s) - \lambda_2| \leq \epsilon/2$ in finite time.

Case 2: $k^{\mathrm{BUD}} < k^{\mathrm{ROS}}$**.** This case is exactly symmetric to Case 1 by flipping μ and λ (see Figure 4b). □

We invoke the following result, which bounds the maximum error between a discrete-time stochastic system and its continuous-time ODE approximation.

Lemma C.2. *Consider the stochastic process $\{Y_t\}_{t \geq 0}$ with $Y_t \in \mathbb{R}^n_{++}$ satisfying*

$$Y_{t+1} = Y_t + \alpha h_t(Y_t)\,,$$

where $h_t : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a random function and $\alpha > 0$ is the step-size. The initial state Y_0 lies in an open subset $\mathcal{Y} \subseteq \mathbb{R}^n$. The random functions are i.i.d. with expectation $\mathbb{E}h_t(y) = \overline{h}(y)$. We assume that the random functions have uniformly bounded expectation $\overline{h}_i(y) \leq \overline{H}$ for all $y \in \mathcal{Y}$, uniformly bounded variance $\mathrm{Var}[h_{t,i}(y)] \leq \overline{H}_2$ for all $y \in \mathcal{Y}$, and its expectation is L-Lipschitz continuous in $\mathcal{Y}$ w.r.t. the max-norm, i.e., $\|\overline{h}(y) - \overline{h}(y')\|_\infty \leq L\|y - y'\|_\infty$ for all $y, y' \in \mathcal{Y}$. Then, the following holds:

1. *The ODE $\frac{d}{ds}\overline{Y}(s) = \overline{h}(\overline{Y}(s))$ with $\overline{Y}(0) = Y_0 \in \mathcal{Y}$ has a unique solution in $\mathcal{Y}$.*
2. *Fix $\epsilon > 0$. Let $\sigma \geq 0$ be such that $\|\overline{Y}(s) - y\|_\infty > \epsilon$ for all $s \in [0, \sigma]$ and $y \notin \mathcal{Y}$. Then,*

$$\mathbb{P}\left\{\max_{t:\alpha t \leq \sigma} \left\|Y_t - \overline{Y}(\alpha t)\right\|_\infty > \epsilon\right\} \leq \epsilon^{-2}\left(\alpha\sigma L\overline{H} + \sqrt{4n\alpha\sigma\overline{H}_2}\right)^2 \exp(2L\sigma)$$

Proof. Denote by $s_t = \alpha t$ the corresponding time in the ODE for the discrete step t. The first part follows from Picard–Lindelöf theorem because $\overline{h}$ is Lipschitz continuous.

We prove the second part in two steps. In the first step, use the Lipschitz continuity of the dynamics to show that deviations of Y_t from the expected path $\overline{Y}(s_t)$ accumulate linearly and conclude by using a discrete version of Gronwall's Lemma to bound the absolute deviations in an almost sure sense. This first step performs a deterministic analysis of the deviations. In the second step, we use a concentration argument to bound the maximum deviation in a stochastic sense.

Step 1. Introduce a time τ corresponding to the first time t with $s_{t+1} \le \sigma$ such that $Y_{t+1} \notin \mathcal{Y}$. Consider a step $t \ge 1$ under the event that $t \le \tau$, which implies that $Y_j \in \mathcal{Y}$ for all $j \le t$ and $\overline{Y}(s) \in \mathcal{Y}$ for all $s \le s_t$. Using the dynamics of the stochastic process and the ODE, we obtain that

$$Y_{t+1} - \overline{Y}(s_{t+1}) = Y_t - \overline{Y}(s(t)) + \alpha h_t(Y_t) - \int_{s_t}^{s_{t+1}} \overline{h}(\overline{Y}(s)) ds \,.$$

From the mean value theorem, because the solution to the ODE is absolutely continuous, we know there exists $\zeta_i \in [s_t, s_{t+1}]$ such that

$$\begin{aligned} \int_{s_t}^{s_{t+1}} \overline{h}_i(\overline{Y}(s)) ds &= (s_{t+1} - s_t) \overline{h}_i(\overline{Y}(\zeta_i)) \\ &= \alpha \overline{h}_i(Y_t) + \underbrace{\alpha \left(\overline{h}_i(\overline{Y}(s_t)) - \overline{h}_i(Y_t) \right) + \alpha \left(\overline{h}_i(\overline{Y}(\zeta_i)) - \overline{h}_i(\overline{Y}(s_t)) \right)}_{\beta_{t,i}} \,. \end{aligned}$$

Therefore, we have that

$$Y_{t+1} - \overline{Y}(s_{t+1}) = Y_t - \overline{Y}(s(t)) + \alpha \Delta_t + \beta_t \,.$$

where $\Delta_t = h_t(Y_t) - \overline{h}(Y_t)$. We refer to Δ_t as a stochastic error and β_t as the integration error. Using that h is L-Lipschitz continuous in $\mathcal{Y}$, the integration error can be bounded as follows:

$$\begin{aligned} |\beta_{t,i}| &\le \alpha \left| \overline{h}_i(\overline{Y}(s_t)) - \overline{h}_i(Y_t) \right| + \alpha \left| (\overline{h}_i(\overline{Y}(\zeta_i)) - \overline{h}_i(\overline{Y}(s_t)) \right| \\ &\le \alpha L \|Y_t - \overline{Y}(s_t)\|_\infty + \alpha L \|\overline{Y}(\zeta_i) - \overline{Y}(s_t)\|_\infty \\ &\le \alpha L \|Y_t - \overline{Y}(s_t)\|_\infty + \alpha^2 L \overline{H} \,, \end{aligned}$$

where the last inequality follows because from the mean value theorem there exists $\zeta_j'' \in [s_t, \zeta_i]$ such that $|\overline{Y}_j(\zeta_i) - \overline{Y}_j(s_t)| = |(\zeta_i - s_t) \overline{h}(Y(\zeta_j''))| \le \alpha \overline{H}$ together with the fact that $|\zeta_i - s_t| \le \alpha$ and $|\overline{h}(y)| \le \overline{H}$.

Therefore, summing over steps $j = 0, \ldots, t$ and using that the initial conditions satisfy $Y_0 = \overline{Y}(0)$, we obtain that the following is true under the event $t \le \tau$:

$$\begin{aligned} \left\| Y_{t+1} - \overline{Y}(s_{t+1}) \right\|_\infty &= \left\| \sum_{j=0}^{t} (\alpha \Delta_j + \beta_j) \right\|_\infty \\ &\le \alpha \|M_t\|_\infty + \alpha L \sum_{j=1}^{t} \|Y_j - \overline{Y}(s_j)\|_\infty + \alpha s_{t+1} L \overline{H} \,, \end{aligned}$$

where the we denote by $M_t = \sum_{j=0}^{t} \Delta_j = \sum_{j=0}^{t} h_j(Y_j) - \overline{h}(Y_j)$ and last inequality follows from the triangle inequality together with $s_t = \alpha t$.

We next apply the following discrete version of Gronwall's Lemma.

Lemma C.3 (Discrete Gronwall's Lemma). *Let $x_t \ge 0$ be a sequence satisfying $x_t \le a + b \sum_{j=1}^{t-1} x_j$ with $a, b \ge 0$. Then, $x_t \le a \exp(bt)$.*

Setting $x_t = \left\| Y_t - \overline{Y}(s_t) \right\|_\infty$ and choosing a, b appropriately, we obtain that

$$\left\| Y_{t+1} - \overline{Y}(s_{t+1}) \right\|_\infty \le \left(\alpha s_{t+1} L \overline{H} + \alpha \max_{\ell = 0, \ldots, t} \|M_\ell\|_\infty \right) \exp(L s_{t+1}) \,.$$

Step 2. Denote by $\mathcal{F}_t = \sigma(h_0, \ldots, h_t)$ the sigma-algebra generated by the random functions up to step t. We have that M_t is a martingale because $M_t \in \mathcal{F}_t$ and $\mathbb{E}[M_{t+1}|\mathcal{F}_t] = M_t$. Moreover, $Y_0, \ldots, Y_{t+1} \in \mathcal{F}_t$ and τ is a stopping time with respect to $\mathcal{F}_t$ because $\tau \in \mathcal{F}_t$.

Taking expectations over the maximum of all steps up to τ, we obtain that

$$\left(\mathbb{E}\left[\max_{j=1,\ldots,\min(t+1,\tau)} \left\|Y_j - \overline{Y}(s_j)\right\|_\infty^2\right]\right)^{1/2} \le \left(\alpha s_{t+1} L\overline{H} + \alpha\left(\mathbb{E}\left[\max_{j=0,\ldots,\min(t,\tau)} \|M_j\|_\infty^2\right]\right)^{1/2}\right)\exp(Ls_{t+1}),$$

where the first inequality follows from Minkowski inequality. It is sufficient to bound each coordinate at a time because

$$\max_{j=0,\ldots,\min(t,\tau)} \|M_j\|_\infty^2 = \max_{i=1,\ldots,n} \max_{j=0,\ldots,\min(t,\tau)} |M_j|^2 \le \sum_{i=1}^n \max_{j=0,\ldots,\min(t,\tau)} |M_{j,i}|^2,$$

where the first equation follows from exchanging maximums and the second since $\|x\|_\infty \le \|x\|_1$. Using that τ is a stopping time and M_t is a martingale that

$$\begin{aligned}
\mathbb{E}\left[\max_{j=0,\ldots,\min(t,\tau)} |M_{j,i}|^2\right] &= \mathbb{E}\left[\max_{j=0,\ldots,t} \left|M_{\min(j,\tau),i}\right|^2\right] \\
&\le 4\mathbb{E}\left[M^2_{\min(t,\tau),i}\right] \\
&= 4\mathbb{E}\left[\left(\sum\nolimits_{j=0}^t \Delta_j \mathbf{1}\{j \le t\}\right)^2\right] = 4\sum_{j=0}^t \mathbb{E}\left[\Delta_j^2 \mathbf{1}\{j \le t\}\right] \\
&\le 4\sum_{j=0}^t \mathbb{E}\left[\Delta_j^2\right] \le 4(t+1)\overline{H}_2,
\end{aligned}$$

where the first inequality follows from Doob's Martingale Inequality because the stopped martingale $M_{\min(t,\tau),i}$ is a martingale, the second equality because martingale differences are orthogonal, and the last our bound on the variance of the random function. Putting everything together, we obtain that

$$\left(\mathbb{E}\left[\max_{j=0,\ldots,\min(t+1,\tau)} \left\|Y_t - \overline{Y}(s_t)\right\|_\infty^2\right]\right)^{1/2} \le \left(\alpha s_{t+1} L\overline{H} + \sqrt{4n\alpha s_{t+1}\overline{H}_2}\right)\exp(Ls_{t+1}). \tag{12}$$

To conclude that if t is the first time win which $\|Y_t - \overline{Y}(s_t)\|_\infty > \epsilon$, then we must have $\|Y_{t-1} - \overline{Y}(s_{t-1})\|_\infty \le \epsilon$, which implies that $Y_{t-1} \in \mathcal{Y}$ (because if $Y_{t-1} \notin \mathcal{Y}$, we would have that $\|Y_{t-1} - \overline{Y}(s_{t-1})\|_\infty$ because $\alpha t - 1 \le \sigma$ and the definition of σ). The latter imples that $\tau \ge t-1$ or $t+1 \le \tau$. Therefore, we can write the event in the statement as

$$\begin{aligned}
\mathbb{P}\left\{\max_{t:\alpha t \le \sigma} \left\|Y_t - \overline{Y}(\alpha t)\right\|_\infty \ge \epsilon\right\} &= \mathbb{P}\left\{\max_{t:\alpha t\le\sigma} \left\|Y_t - \overline{Y}(\alpha t)\right\|_\infty \mathbf{1}\{t+1 \le \tau\} \ge \epsilon\right\} \\
&= \mathbb{P}\left\{\max_{t:\alpha t \le \sigma, t \le \tau - 1} \left\|Y_t - \overline{Y}(\alpha t)\right\|_\infty \ge \epsilon\right\} \\
&\le \epsilon^{-2}\mathbb{E}\left[\max_{t:\alpha t\le\sigma, t\le\tau-1} \left\|Y_t - \overline{Y}(\alpha t)\right\|_\infty^2\right] \\
&\le \epsilon^{-2}\left(\alpha s_t L\overline{H} + \sqrt{4n\alpha s_t \overline{H}_2}\right)^2 \exp(2Ls_t),
\end{aligned}$$

where the first inequality follows from an application of Markov's inequality and the last from (12). We conclude by noting that $s_t \le \sigma$. □

We apply Lemma C.2 to $Y_t = (\log\lambda_t, \log\mu_t)$ and set the random function h_t to be the gradients of the ROS and budget constraints, respectively. That is,

$$h_t(y) = h_t(\log\lambda, \log\mu) = -\left(v_t \cdot x_t\left(v_t \cdot k^{\min}(\lambda,\mu)\right) - p_t\left(v_t \cdot k^{\min}(\lambda,\mu)\right), \rho - p_t\left(v_t \cdot k^{\min}(\lambda,\mu)\right)\right).$$

This choice reduces the stochastic process in the statement of the lemma to the update rule of the algorithm. Taking expectations, we obtain that

$$\overline{h}(\log\lambda, \log\mu) = -\left(g^{\text{ROS}}\left(k^{\min}(\lambda,\mu)\right), g^{\text{BUD}}\left(k^{\min}(\lambda,\mu)\right)\right)$$

By assumption, the expected gradients are bounded and have finite variance. For Lipschitz continuity we need to show that for $g = g^{\text{BUD}}, g^{\text{ROS}}$

$$\left|g(k^{\min}(\lambda,\mu)) - g(k^{\min}(\lambda',\mu'))\right| \leq L\max\left(\|\log\lambda - \log\lambda'\|, \|\log\mu - \log\mu'\|\right).$$

The expected gradients, however, are not Lipschitz continuous for all multipliers because of the logarithmic transformation. To guarantee Lipschitz continuity, we restrict the set of dual solutions to lie in the set

$$\mathcal{Y} = \left\{(\log(\lambda), \log(\mu)) \in \mathbb{R}^2 : \lambda > \epsilon/2 \text{ or } \mu > \epsilon/2\right\}.$$

For example, the gradient of the budget constraint be written as

$$\begin{aligned} g^{\text{BUD}}\left(k^{\min}(\lambda,\mu)\right) &= g^{\text{BUD}}\left(k^{\min}(\exp(\log(\lambda), \exp(\log(\mu))\right) \\ &= g^{\text{BUD}}\left(\min\left(\exp(-\log(\lambda)) + 1\right), \exp(-\log(\mu))\right) \\ &= g^{\text{BUD}}\left(\exp\min\left(\log(\exp(-\log(\lambda)) + 1), -\log(\mu)\right)\right). \end{aligned}$$

Because the minumum $\min(x,y)$ and the log-sum-exp function $\log(\exp(x)+1)$ are 1-Lipschitz continuous, we obtain that

$$\min\left(\log(\exp(-\log(\lambda)) + 1), -\log(\mu)\right)$$

is 1-Lipschitz continuous in $(\log\lambda, \log\mu)$. For $(\log\lambda, \log\mu) \in \mathcal{Y}$ we have that $k^{\min}(\lambda,\mu) < 2/\epsilon$, which implies that $g^{\text{BUD}}\left(k^{\min}(\lambda,\mu)\right)$ is $2L_g/\epsilon$-Lipscthiz continuous because the exponential $\exp(x)$ function is $\exp(a)$-Lipschitz continuous in $[0,a]$.

In Lemma C.2, we set σ as the time it takes the ODE to reach the set $\mathcal{O}_{\epsilon/2}$ and $\epsilon := \epsilon/2$. Under the good event $A = \left\{\max_{t:\alpha t\leq\sigma} \left\|Y_t - \overline{Y}(\alpha t)\right\|_\infty \leq \epsilon/2\right\}$, we have by Lemma C.1 that $(\lambda_t, \mu_t) \notin [0,\epsilon]^2$ and, thus, the dynamics are Lipschitz continuous. Moreover, we because the step size is $\alpha \approx T^{-1/2}$ we have that at time $\tau = \lfloor\sigma/\alpha\rfloor = O\left(T^{1/2}\right)$ the state of the algorithm reaches the orbit $\mathcal{O}_\epsilon$ with high probability. More formally, we have proved the following result.

Proposition C.4. *For every initial dual solution $(\lambda_1, \mu_1) \notin [0,\epsilon)^2$, there exists a time $\tau = O\left(T^{1/2}\right)$ such that the probability of not hitting the orbit is bounded by*

$$\mathbb{P}\left\{(\lambda_\tau, \mu_\tau) \notin \mathcal{O}_\epsilon\right\} = O\left(T^{-1/2}\right).$$

C.3.2. STEP 2: ORBITAL STABILITY

We next show that once the iterates reach the orbit $\mathcal{O}_\epsilon$, they stay in the orbit for the rest of the horizon with high probability. To prove this result we show that the sum of Bregman divergence V_h induced by the negative entropy $h(u) = u\log u$ constitutes a stochastic Lyaponuv function. The Lyaponuv function is given by

$$V(\lambda,\mu) = V_h(\lambda^*, \lambda) + V_h(\mu^*, \mu),$$

where the Bregman divergence $V_h(y,x) = h(y) - h(x) - h'(x)\cdot(y-x)$ is $V_h(y,x) = y\log(y/x) - y + x$. Note that we can choose $m > 0$ such that $V(\lambda,\mu) < m$ for all $(\lambda,\mu) \in \mathcal{O}_\epsilon$.

Assume that the ROS constraint is binding so that $\mu^* = 0$. Here, we have that $V_h(\mu^*, \mu) = \mu$. Let

$$g_t^{\text{ROS}} = v_t \cdot x_t\left(v_t \cdot k^{\min}(\lambda_t,\mu_t)\right) - p_t\left(v_t \cdot k^{\min}(\lambda_t,\mu_t)\right) \quad \text{and} \quad g_t^{\text{BUD}} = \rho - p_t\left(v_t \cdot k^{\min}(\lambda_t,\mu_t)\right)$$

be the empirical gradients at time t. The multiplicative weight update implies that

$$V(\lambda_{t+1},\mu_{t+1}) \leq V(\lambda_t,\mu_t) - \alpha g_t^{\text{ROS}}\cdot(\lambda_t - \lambda^*) - \alpha g_t^{\text{BUD}}\cdot(\mu_t - \mu^*) + \alpha^2\frac{\overline{G_2}}{\lambda^* - \epsilon},$$

where we used that second moments of the gradients are bounded by Assumption 3.5 and that the Bregman divergence is $(\lambda^* - \epsilon)^{-1}$-locally-strongly-convex in $\mathcal{O}_\epsilon$ by Lemma 2. Taking expectations conditional on the current iterates, we obtain that

$$\mathbb{E}\left[V_h(\lambda_{t+1}, \mu_{t+1}) \mid \lambda_t, \mu_t\right] \leq V_h(\lambda_t, \mu_t) - \alpha g^{\mathrm{ROS}}\left(k^{\min}(\lambda_t, \mu_t)\right) \cdot (\lambda_t - \lambda^*) - \alpha g^{\mathrm{BUD}}\left(k^{\min}(\lambda_t, \mu_t)\right) \cdot (\mu_t - \mu^*) + \alpha^2 \frac{\overline{G_2}}{\lambda^* - \epsilon}\,.$$

For the budget constraint, we know that in the set $\mathcal{O}_\epsilon$ there exists $\underline{g} > 0$ such that $g^{\mathrm{BUD}}\left(k^{\min}(\lambda, \mu)\right) \geq \underline{g}$. Therefore, using that $\mu^* = 0$ and $\mu_t \geq 0$ we obtain that

$$g^{\mathrm{BUD}}\left(k^{\min}(\lambda_t, \mu_t)\right) \cdot (\mu_t - \mu^*) = g^{\mathrm{BUD}}\left(k^{\min}(\lambda_t, \mu_t)\right) \cdot \mu_t \geq \underline{g} \cdot \mu_t \geq \underline{g} \cdot V_h(\mu^*, \mu_t)\,.$$

For the ROS constraint, use that $k^* = 1/\lambda^* + 1$ and $k_t := k^{\min}(\lambda_t, \mu_t) = (1 + \lambda_t)/\lambda_t$ for $(\lambda_t, \mu_t) \in \mathcal{O}_\epsilon$ to obtain that

$$\begin{aligned} g^{\mathrm{ROS}}\left(k^{\min}(\lambda_t, \mu_t)\right) \cdot (\lambda_t - \lambda^*) &= -g^{\mathrm{ROS}}(k_t) \cdot (k_t - k^*) \cdot \frac{\lambda^* - \lambda_t}{k_t - k^*} = -g^{\mathrm{ROS}}(k_t) \cdot (k_t - k^*) \cdot \lambda_t \cdot \lambda^* \\ &\geq \ell (k_t - k^*)^2 \cdot \lambda_t \cdot \lambda^* = \frac{\ell}{\lambda_t \cdot \lambda^*}(\lambda_t - \lambda^*)^2 \\ &\geq \frac{\ell}{\lambda^*(\lambda^* + \epsilon)} \cdot (\lambda - \lambda^*)^2 \\ &\geq \frac{2\ell(\lambda^* - \epsilon)^2}{(\lambda^*)^2(\lambda^* + \epsilon)} \cdot V_h(\lambda^*, \lambda)\,, \end{aligned}$$

where the first inequality follows from the strong monotonicity condition in Assumption 3.6 and that dual variables are non-negative, the second inequality because $\lambda_t \leq \lambda^* + \epsilon$ for all $(\lambda_t, \mu_t) \in \mathcal{O}_\epsilon$, and the last inequality follows because the Bregman divergence of the negative entropy function satisfies $V_h(y, x) \leq y/(2\min(x, y)^2)(y - x)^2$ for $x, y > 0$ together with $\lambda_t \geq \lambda^* - \epsilon$ for all $(\lambda_t, \mu_t) \in \mathcal{O}_\epsilon$. Putting everything together, we obtain that there exists constant C_1, C_2 such that

$$\mathbb{E}\left[V(\lambda_{t+1}, \mu_{t+1}) \mid \lambda_t, \mu_t\right] \leq (1 - \alpha C_1) V(\lambda_t, \mu_t) + \alpha^2 C_2\,.$$

We are now ready to invoke the following classical theorem on stochastic stability.

Theorem C.5 (Kushner (1967, p. 86))**.** *Let $x_t, t = 1, \ldots, T$ be a Markov process and $V(x)$ a continuous non-negative function with*

$$\mathbb{E}\left[V(x_{t+1}) \mid x_t\right] \leq V(x_t)/\beta + \phi$$

for every x such that $V(x) < m$, where $\beta > 1$ and $\phi \geq 0$. Then

$$\mathbb{P}\left\{\max_{t=1,\ldots,T} V(x_t) \geq m\right\} \leq \frac{V(x_1)}{\beta^T m} + \frac{(1 - \beta^{-T})\phi\beta}{(\beta - 1)m}\,.$$

Setting $\beta = 1/(1 - C_1\alpha)$ and $\phi = C_2\alpha^2$, we obtain that for $C_1\alpha < 1$, which holds for large enough T

$$\begin{aligned} \mathbb{P}\left\{\exists t : (\lambda_t, \mu_t) \notin \mathcal{O}_\epsilon, \tau < t \leq T \mid (\lambda_\tau, \mu_\tau) \in \mathcal{O}_\epsilon\right\} &\leq \mathbb{P}\left\{\max_{t=\tau+1,\ldots,T} V(\lambda_t, \mu_t) \geq m \mid (\lambda_\tau, \mu_\tau) \in \mathcal{O}_\epsilon\right\} \\ &\leq \beta^{-T} + \frac{(1 - \beta^{-T})\phi\beta}{(\beta - 1)m} \\ &\leq \exp(-C_1 T\alpha) + \frac{C_2\alpha}{C_1 m}\,, \end{aligned}$$

where the last equation follows because $\phi\beta/(\beta - 1) = C_2\alpha/C_1$, $\beta^{-T} = (1 - C_1\alpha)^T \leq \exp(-C_1 T\alpha)$. Setting $\alpha \approx T^{-1/2}$ we obtain the following result.

Proposition C.6. *The algorithm is orbital stable, that is, the probability of leaving the orbit after time τ is bounded by*

$$\mathbb{P}\left\{\exists t : (\lambda_t, \mu_t) \notin \mathcal{O}_\epsilon, \tau < t \leq T \mid (\lambda_\tau, \mu_\tau) \in \mathcal{O}_\epsilon\right\} = \left(T^{-1/2}\right)\,.$$

C.3.3. Step 3: Regret Analysis

As before, suppose that the ROS constraint is binding at the optimal solution. Consider an alternate algorithm that (1) behaves as the original algorithm up to time τ and (2) after time τ always uses the multiplier of the ROS constraint and projects the dual variable to $[0, \lambda^* + \epsilon]$. Let $\widehat{\lambda}_t$ be the dual variable in this new algorithm and denote the multiplier used by $\widehat{k}_t = (1 + \widehat{\lambda}_t)/\widehat{\lambda}_t$. We denote by $E_t = \left\{k^{\min}(\lambda_t, \mu_t) = \widehat{k}_t\right\}$ the event that the multipliers used by both algorithm match.

Let τ^{BUD} be a stopping time as defined in Definition 3.1 corresponding to the first time the budget is depleted for some initial budget $B = \rho T$. Because values are non-negative, we can lower bound the reward of MinPacing given any $\overrightarrow{\gamma}$ by summing over the value collected only in iterations from τ up to τ^{BUD} and conditioning on the event E_t

$$
\begin{aligned}
\mathsf{Reward}(\mathsf{MinPacing}, \overrightarrow{\gamma}, \rho) &\geq \sum_{t=\tau}^{\tau^{\mathrm{BUD}}} v_t \cdot x_t \left(v_t \cdot k^{\min}(\lambda_t, \mu_t)\right) \cdot E_t \\
&= \sum_{t=\tau}^{\tau^{\mathrm{BUD}}} v_t \cdot x_t \left(v_t \cdot (1 + \widehat{\lambda}_t)/\widehat{\lambda}_t\right) \cdot E_t \\
&\geq \underbrace{\sum_{t=\tau}^{\tau^{\mathrm{BUD}}} v_t \cdot x_t \left(v_t \cdot (1 + \widehat{\lambda}_t)/\widehat{\lambda}_t\right)}_{(I)} - \underbrace{\sum_{t=\tau}^{T} v_t \cdot (1 - E_t)}_{(II)}
\end{aligned}
$$

where the first equation follows from the definition of the event E_t, and the last inequality follows because $x_t \leq 1$ and adding back periods after τ^{BUD}. We bound each term at a time.

For the first term, use that the alternate algorithm always bid according to the ROS constraint to write

$$
v_t \cdot x_t \left(v_t \cdot (1 + \widehat{\lambda}_t)/\widehat{\lambda}_t\right) = 0 \cdot \rho + f_t^{\star}(\widehat{\lambda}_t, 0) - \sum_{t=\tau}^{\tau^{\mathrm{BUD}}} \widehat{\lambda}_t \cdot g_t^{\mathrm{ROS}}(\widehat{\lambda}_t, 0)
$$

Taking expectations, we can use that τ^{BUD} is a stopping time and a martingale argument to obtain that

$$
\mathbb{E}_{\overrightarrow{\gamma}_T \sim \mathcal{P}^T}\left[(I)\right] \geq \mathbb{E}_{\overrightarrow{\gamma}_T \sim \mathcal{P}^T}\left[\sum_{t=\tau}^{\tau^{\mathrm{BUD}}} D(\widehat{\lambda}_t, 0) - \sum_{t=\tau}^{\tau^{\mathrm{BUD}}} \widehat{\lambda}_t \cdot g_t^{\mathrm{ROS}}(\widehat{\lambda}_t, 0)\right],
$$

where $D(\lambda, \mu)$ is the dual function. Let $\overline{\lambda} = (\tau^{\mathrm{BUD}} + 1 - \tau)^{-1} \sum_{t=\tau}^{\tau^{\mathrm{BUD}}} \widehat{\lambda}_t$ be the average dual variable for the ROS constraint. Using the convexity of the dual function we obtain that

$$
\sum_{t=\tau}^{\tau^{\mathrm{BUD}}} D(\widehat{\lambda}_t, 0) \geq (\tau^{\mathrm{BUD}} + 1 - \tau) D(\overline{\lambda}, 0) \geq \mathbb{E}_{\overrightarrow{\gamma} \sim \mathcal{P}^T}\left[\mathsf{Reward}(\mathsf{OPT}, \overrightarrow{\gamma})\right] - O\left(T^{1/2}\right),
$$

where we used that $(\overline{\lambda}, 0)$ is dual feasible and weak duality together with $\tau = O\left(T^{1/2}\right)$ and $T - \tau^{\mathrm{BUD}} = O\left(T^{1/2}\right)$. Because the alternate algorithm projects dual variables to $[0, \lambda^* + \epsilon]$, Lemma 2 implies that the Bregman divergence of the generalized negative entropy is $1/(\lambda^* + \epsilon)$-strongly convex. Applying the mirror descent guarantee in Lemma 1 to the linear functions $w_t(\lambda) = \lambda \cdot g_t^{\mathrm{ROS}}(\widehat{\lambda}_t, 0)$ we obtain that

$$
\sum_{t=\tau}^{\tau^{\mathrm{BUD}}} \widehat{\lambda}_t \cdot g_t^{\mathrm{ROS}}(\widehat{\lambda}_t, 0) = \sum_{t=\tau}^{\tau^{\mathrm{BUD}}} w_t(\widehat{\lambda}_t) - w_t(0) = O\left(T^{1/2}\right),
$$

because the alternate algorithm updates the dual variable of the ROS constraint according to $g_t^{\mathrm{ROS}}(\widehat{\lambda}_t, 0)$. Therefore, we have that

$$
\mathbb{E}_{\overrightarrow{\gamma}_T \sim \mathcal{P}^T}\left[(I)\right] \geq \mathbb{E}_{\overrightarrow{\gamma} \sim \mathcal{P}^T}\left[\mathsf{Reward}(\mathsf{OPT}, \overrightarrow{\gamma})\right] - O\left(T^{1/2}\right). \tag{13}
$$

For the second term, using that values are independent of the event E_t we obtain

$$\begin{aligned}
\mathbb{E}_{\vec{\gamma}_T\sim\mathcal{P}^T}\left[(II)\right] &= \sum_{t=\tau}^{T}\mathbb{E}[v_t]\cdot\mathbb{P}\left\{E_t^{\complement}\right\} \\
&\leq T\cdot\mathbb{E}[v]\cdot\mathbb{P}\left\{\cup_{t=\tau}^{T}E_t^{\complement}\right\} \\
&= T\cdot\mathbb{E}[v]\cdot\left(\mathbb{P}\left\{\cup_{t=\tau}^{T}E_t^{\complement}\mid A\right\}\mathbb{P}\left\{A\right\}+\mathbb{P}\left\{\cup_{t=\tau}^{T}E_t^{\complement}\mid A^{\complement}\right\}\mathbb{P}\left\{A^{\complement}\right\}\right) \\
&\leq T\cdot\mathbb{E}[v]\cdot\left(\mathbb{P}\left\{\exists t:(\lambda_t,\mu_t)\notin\mathcal{O}_\epsilon,\tau<t\leq T\mid(\lambda_\tau,\mu_\tau)\in\mathcal{O}_\epsilon\right\}+\mathbb{P}\left\{(\lambda_\tau,\mu_\tau)\notin\mathcal{O}_\epsilon\right\}\right) \\
&= O\left(T^{1/2}\right), \qquad (14)
\end{aligned}$$

where the first inequality follows because values are i.i.d. and $\mathbb{P}\{E_t^{\complement}\}\leq\mathbb{P}\{\cup_{t=\tau}^{T}E_t^{\complement}\}$ for all $t=\tau,\ldots,T$, the second equality follows from conditioning on the event $A=\{(\lambda_\tau,\mu_\tau)\in\mathcal{O}_\epsilon\}$, the second inequality follows because probabilities are at most one and if for some t the event $E_t^{\complement}$ is true then it must be the case that $(\lambda_t,\mu_t)\notin\mathcal{O}_\epsilon$ since the algorithm bids according to the ROS multiplier in the orbit of the optimal dual solution, and the last inequality follows from Proposition C.4 and Proposition C.6.

Combining (13) and (14) we conclude that

$$\mathsf{Regret}(\mathsf{MinPacing},\mathcal{P}^T)=O\left(T^{1/2}\right).$$

C.3.4. Online Mirror Descent Results

The following are some known results of Online Mirror Descent that we used in our previous analysis.

Lemma 1 (Bubeck et al. 2015, Theorem 4.2)**.** *Let h be a mirror map which is ρ-strongly convex on $\mathcal{X}\cap\mathcal{D}$ with respect to a norm $\|\cdot\|$. Let f be convex and L-Lipschitz with respect to $\|\cdot\|$. Then, mirror descent with step size α satisfies*

$$\sum_{s=1}^{t}\left(f(x_s)-f(x)\right)\leq\frac{1}{\alpha}V_h(x,x_1)+\alpha\frac{L^2t}{2\rho}.$$

Lemma 2 (Allen-Zhu and Orecchia 2014)**.** *The Bregman divergence of the generalized negative entropy satisfies "local strong convexity": for any $x,y>0$,*

$$V_h(y,x)=y\log(y/x)+x-y\geq\frac{1}{2\max(x,y)}\cdot(y-x)^2.$$

Proof. The claimed inequality is equivalent to

$$t\log t\geq(t-1)+\frac{1}{2\max(1,t)}\cdot(t-1)^2 \qquad (15)$$

for $t>0$. Suppose $t\geq1$. Then, choosing $u=1-1/t$, Inequality (15) is equivalent to

$$-\log(1-u)\geq u+\frac{1}{2}u^2,$$

for $u\in[0,1)$, which holds by Taylor series. Suppose $0<t\leq1$. Then Inequality (15) is equivalent to

$$\log t-\frac{1}{2}\left(t-\frac{1}{t}\right)\geq0,$$

which may be checked by observing that the function is decreasing and equals zero at $t=1$. This completes the proof of the claim. □

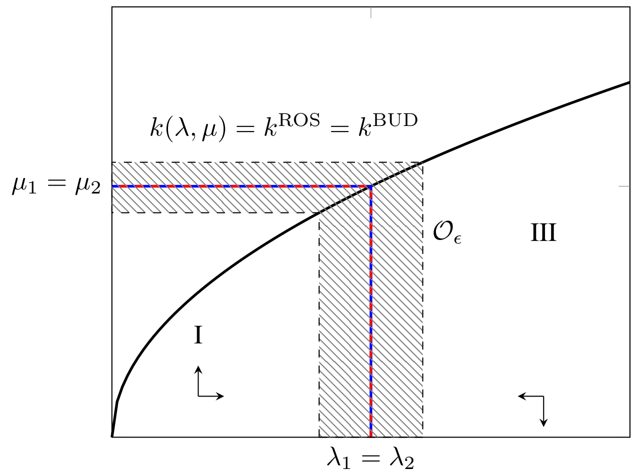


Figure 5. Illustration of the MIN dynamics in degenerate case ($k^{\mathrm{BUD}} = k^{\mathrm{ROS}}$). The alternating dashed blue and red line indicates the set of all optimal solutions, i.e., the set of dual variables for which $k^{\min}(\lambda, \mu) = k^{\mathrm{ROS}} = k^{\mathrm{BUD}}$. The hatched rectangle is an orbit $\mathcal{O}_\epsilon$ of size ϵ around the set optimal solutions. The solid black curve gives the points for which the $(1+\lambda)/\lambda = 1/\mu$, i.e., the multipliers of both constraints are equal. Above the curve, the algorithm bids $1/\mu$ according to the budget constraint, and below it bids $(1+\lambda)/\lambda$ according to the ROS constraint. The arrows indicate the drift of the stochastic process in each region.

C.4. Analysis in the Degenerate Case

Without Assumption 3.4 we have multiple optimal dual solutions. In particular, the optimal solutions are all (λ, μ) with $\min(1 + 1/\lambda, 1/\mu) = k^*$. Visually, the red and blue curves in Figure 4 would coincide, and all points along these curves would be optimal. As a result, the orbit $\mathcal{O}_\epsilon$ would not be a ball around the unique optimal solution but a band around this curve of all optimal solutions. The dynamics for the degenerate case are illustrated in Figure 5. We can still follow similar steps as in the proof of Theorem 1.

First, we can show that it takes order $\sqrt{T}$ steps to get to the orbit of all optimal solutions using a similar ODE analysis. This can be easily seen in Figure 5 as the drift in regions (I) and (III) point toward the orbit $\mathcal{O}_\epsilon$. Second, we can prove orbital stability, i.e., once the algorithm reaches the orbit, it never leaves with a high probability. Third, we can show that the regret accumulated once the algorithm is in the orbit of the optimal solution is $O(\sqrt{T})$.

C.5. Proof of Theorem 2

To prove the result, we first show the next lemma, which bounds the constraint violations for time t.

Lemma 3. *Recall $g_t^{\mathrm{ROS}} = v_t \cdot x_t(b_t) - p_t(b_t)$ and $g_t^{\mathrm{BUD}} = \rho - p_t(b_t)$ with λ_t, μ_t being the dual variables for the ROS and the budget constraint respectively, and $b_t = v_t \cdot k^{\min}(\lambda_t, \mu_t) = v_t \cdot \min\{1/\mu_t, 1 + 1/\lambda_t\}$ being the bid used by the algorithm. If the payment and allocation functions satisfy $0 \le p_t(b) \le b \cdot x_t(b_t)$ for any bid $b > 0$ (e.g. truthful auctions), then we have*

$$g_t^{\mathrm{ROS}} \ge -\frac{1}{\lambda_t} \quad \textit{and} \quad g_t^{\mathrm{BUD}} \ge \rho - \frac{1}{\mu_t}$$

Proof. Our condition only says the payment is always non-negative and at most the bid. Recall we also normalize the functions so that v_t, p_t and x_t all have range $[0, 1]$. For the ROS constraint, $b_t \le v_t \cdot (1 + 1/\lambda_t)$ and $p_t(b_t) \le b_t \cdot x_t(b_t)$, we get

$$g_t^{\mathrm{ROS}} \ge (v_t - b_t) \cdot x_t(b_t) \ge -\frac{v_t}{\lambda_t} \cdot x_t(b_t) \ge -\frac{1}{\lambda_t}.$$

Similarly, for the budget constraint because $b_t \le v_t/\mu_t$, we get

$$g_t^{\mathrm{BUD}} \ge \rho - b_t \cdot x_t(b_t) \ge \rho - \frac{v_t}{\mu_t} \cdot x_t(b_t) \ge \rho - \frac{1}{\mu_t}.$$

□

The first result in Theorem 2 on ROS constraint violation can be obtained from the below lemma.

Lemma 4. *Consider a run of the min pacing algorithm starting at $\lambda_1 > 0$ and $\alpha = \frac{1}{\sqrt{T}}$, then for any outcome $\vec{\gamma}$ over the T iterations, the ROS constraint violation satisfies*

$$\sum_{t=1}^{T} p_t(b_t) - v_t \cdot x_t(b_t) = -\sum_{t=1}^{T} g_t^{\mathrm{ROS}} \leq 2\sqrt{T} \log \frac{T}{\lambda_1}.$$

Proof. Equation (4) in the algorithm implies $\lambda_{t+1} = \exp\left[-\alpha \sum_{t'=1}^{t} g_{t'}^{\mathrm{ROS}}\right]$. If $-\sum_{t=1}^{T} g_t^{\mathrm{ROS}} \leq \sqrt{T} \log \frac{T}{\lambda_1}$, we are done. Otherwise, let T' be the last time that $-\sum_{t=1}^{T'} g_t^{\mathrm{ROS}} \leq \sqrt{T} \log \frac{T}{\lambda_1}$, so we know for any $t > T'$, the dual variable λ_t must be larger than T since

$$\lambda_t = \lambda_1 \cdot \exp\left[-\alpha \sum_{t'=1}^{t} g_{t'}^{\mathrm{ROS}}\right] > \lambda_1 \cdot \exp\left[\alpha\sqrt{T} \log \frac{T}{\lambda_1}\right] = T$$

By Lemma 3 we know $g_t^{\mathrm{ROS}} \geq -\frac{1}{\lambda_t}$, so $-g_t^{\mathrm{ROS}} \leq \frac{1}{\lambda_t} \leq \frac{1}{T}$ for all the iterations t after T'. Since there are at most T such iterations, we get

$$-\sum_{t=1}^{T} g_t^{\mathrm{ROS}} = -\sum_{t=1}^{T'} g_t^{\mathrm{ROS}} - \sum_{t>T'} g_t^{\mathrm{ROS}} \leq \sqrt{T} \log \frac{T}{\lambda_1} + 1 \leq 2\sqrt{T} \log \frac{T}{\lambda_1}.$$

□

The second result in Theorem 2 on stopping time can be obtained from the below lemma.

Lemma 5. *Let $\mu^{\max} = 1/\rho + 1$, and consider a run of the min pacing algorithm starting at $\mu_1 \in (0, \mu^{\max}]$ and $\eta = \frac{1}{\sqrt{T}}$, then for any outcome $\vec{\gamma}$ over the T iterations, we have $\mu_t \leq \mu^{\max}$ for all $t \leq \tau^{\mathrm{BUD}}$, and*
$T - \tau^{\mathrm{BUD}} \leq \frac{\sqrt{T}}{\rho} \cdot \log \frac{10\mu^{\max}}{\mu_1} = O(\sqrt{T})$

Proof. The part of $\mu_t \leq \mu^{\max}$ follows inductively. If $\mu_t \leq \mu^{\max}$, either $\mu_t \leq 1/\rho$, then since the step-size η is chosen to be small enough we have $\mu_{t+1} \leq 1/\rho + 1$, otherwise if $\mu_t > 1/\rho$, by Lemma 3 we know $g_t^{\mathrm{BUD}} > 0$ and thus $\mu_{t+1} \leq \mu_t \leq \mu^{\max}$.

The part of τ^{BUD} can be shown by contradiction. Suppose $\tau^{\mathrm{BUD}} < T - \frac{\sqrt{T}}{\rho} \cdot \log \frac{10\mu^{\max}}{\mu_1}$, it means $\sum_{t=1}^{\tau^{\mathrm{BUD}}-1} p_t(b_t) \geq \rho \cdot T - 2$ and thus

$$\sum_{t=1}^{\tau^{\mathrm{BUD}}-1} g_t^{\mathrm{BUD}} \leq \rho \cdot \tau^{\mathrm{BUD}} - (\rho \cdot T - 2) \leq -\sqrt{T} \cdot \log \frac{10\mu^{\max}}{\mu_1} + 2.$$

Similar to the ROS case, note $\mu_{\tau^{\mathrm{BUD}}} = \mu_1 \cdot \exp\left[-\eta \sum_{t=1}^{\tau^{\mathrm{BUD}}-1} g_t^{\mathrm{BUD}}\right] \geq \mu^{max}$, which gives a contradiction. □

D. Proof of Proposition 3.7: Analysis of Sequential Algorithm

We prove Proposition 3.7 in this section. That is, we will show that for any initialization of the sequential pacing algorithm, i.e. choice of initial values μ_0, λ_0 of the dual variables and their respective step-sizes η, α, there will always be some instance on which the algorithm performs poorly, i.e. it either violates the ROS constraint by at least $\Omega(T)$ or has a regret at least $\Omega(T)$.

Without loss of generality, we assume μ_0 and λ_0 are both $O(1)$. All the instances we use in the proof will be deterministic, i.e. $v, x(\cdot), p(\cdot)$ are drawn i.i.d from a point distribution. In particular, all instances we consider have fixed values $v_t = 1$, $x_t(b) = \min(\frac{b}{4}, 1)$ and $p_t(b) = \min(\frac{b^2}{8}, 2)$ for all $b \geq 0$. Effectively the bid ranges from 0 to 4, and is equivalent to the bid multiplier as $v = 1$. We pick these values for notation simplicity, and it is easy to scale all quantities down to satisfy our model where v, x, p are all in $[0, 1]$. Note that the payment function p is the truthful pricing corresponding to the allocation function x in our example. We start with the following observations for our instance.

Observation D.1. *It is straightforward to see that in each iteration, the value is a concave function on the payment, i.e.* $(v \cdot x) = \sqrt{p/2}$ *(Figure6), and thus if we fix some total spend* P *over some* t *iterations, the largest total value is achieved by spending evenly (i.e.* P/t*) in each of the* t *iterations. Similarly because of concavity, if there is an additional constraint that the per-iteration spend is at least* $l \geq P/t$*, the optimal total value is achieved by spending* l *per-iteration (over any* $P/l < t$ *iterations).*

Observation D.2. *In each iteration, the largest ROS slack one can achieve is at most* $1/8$*, i.e.,* $\max_b \{v \cdot x(b) - p(b)\} = 1/8$ *by bidding* $b = 1$.

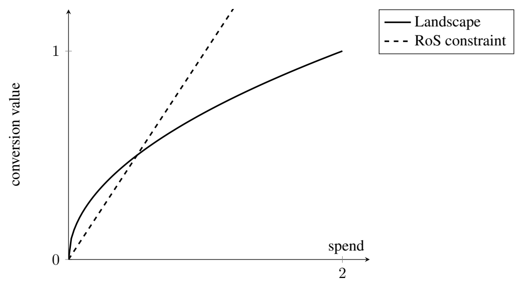


Figure 6. Achievable spend vs conversion value for the sequential example.

Fix any $\mu_0, \lambda_0, \eta, \alpha$, we will consider a pair of instances. The first instance $\widehat{\mathcal{I}}$ has $\widehat{\rho} = 1.9$ (and the v, x, p as described above). It is easy to see for $\widehat{\mathcal{I}}$ that $k^* = k^{\text{ROS}} = 2$ and $k^{\text{BUD}} = \sqrt{8 \cdot 1.9} > 2$, and it is optimal to spend 0.5 per iteration and get $T \cdot v \cdot \frac{2}{4} = T/2$ total value. It is also easy to check that this instance satisfies all the assumptions we need for the min-pacing algorithm. Consider the sequential pacing algorithm with two cases

1. If the total spend over T iterations is at least $P \geq 0.6 \cdot T$. The maximum total value in this case is achieved by spending P/T per iteration (Observation D.1), which means bidding $b = \sqrt{8P/T}$ and get value $\sqrt{P/(2T)}$. Thus the total value is at most $\sqrt{P \cdot T/2}$, so the total ROS constraint violation is at least

$$P - \sqrt{P \cdot T/2} = (P/T - \sqrt{\frac{P}{2T}}) \cdot T.$$

   It is easy to check this is at least $\Omega(T)$ when $P \geq 0.6 \cdot T$, so the ROS constraint violation would be linear in T.

2. If the total spend over T iterations is at most $0.6 \cdot T$. Consider any iteration after the first $0.7 \cdot T$ iterations, and we know $\mu_t = \mu_0 \exp\Big(- \eta \cdot (\widehat{\rho} \cdot t - \sum_{t'<t} p_{t'}(b_{t'})\Big) \leq \mu_0 \exp\Big(- \eta \cdot 0.73 \cdot T\Big)$ for any $t \geq 0.7T$ since $\widehat{\rho} = 1.9$ and total spend is at most $0.6 \cdot T$. There are two sub-cases:

   - If $\mu_0 \exp\Big(- \eta \cdot 0.73 \cdot T\Big) \leq 1/3$, we know $\mu_t \leq 1/3$ and thus $b_t = \frac{1}{\mu_t} \cdot \frac{\lambda_t + 1}{\lambda_t} \geq 3$ for all $t \geq 0.7 \cdot T$, which means we will have a per iteration ROS violation of at least $9/8 - 3/4 = 0.375$ (with $b_t = 3$) in each of the last $0.3 \cdot T$ iterations, since the ROS violation increases with b over the region $b \geq 3$. In each of the first $0.7 \cdot T$ iterations, the ROS slack we can gain is at most $1/8$ (Observation D.2), so the total ROS constraint violation is at least $0.375 \cdot 0.3 \cdot T - 0.7 \cdot T/8 \geq 0.025 \cdot T$.
   - If $\mu_0 \exp\Big(- \eta \cdot 0.73 \cdot T\Big) > 1/3$, then we know $\mu_0 > 1/3$ and $\eta \leq \frac{\ln(3\mu_0)}{0.73 \cdot T}$.

We can conclude from the above discussion that the only possible scenario where an instantiation of the sequential pacing algorithm won't incur a $\Omega(T)$ violation of the ROS constraint on the instance $\widehat{\mathcal{I}}$ is in the last sub-case, which means the step-size η of the budget dual variable is $O(1/T)$. If that is the case, it is easy to see such an instantiation must perform poorly on a budget-binding instance when we need η to be large so the budget dual variable μ can increase fast enough to lower the bid sufficiently.

More specifically, when $\mu_0 \exp\Big(- \eta \cdot 0.73 \cdot T\Big) > 1/3$ holds, we consider the instance $\widetilde{\mathcal{I}}$ with $\widetilde{\rho} = \frac{1}{200\mu_0^4}$. Note $\widetilde{\rho}$ is $\Theta(1)$ since we assume μ_0 is $O(1)$ and in this case $\mu_0 > 1/3$. We have $k^* = k^{\text{BUD}} = \frac{1}{5\mu_0^2} < 1.8$, $k^{\text{ROS}} = 2$, and the maximum total value is $\frac{T}{20\mu_0^2}$ achieved by bidding k^{BUD} and spending $\widetilde{\rho}$ in each iteration. Since the pacing algorithm guarantees to obey the budget constraint (by not bidding above the remaining budget at any time), the total spend is at most $\widetilde{\rho} \cdot T = \frac{T}{200\mu_0^4} \leq 0.73 \cdot T$ (as $\mu_0 > 1/3$), so we know that $\sum_{t' \leq t} \widetilde{\rho} - p_{t'}(b_{t'}) \geq -\sum_{t' \leq t} p_{t'}(b_{t'}) \geq -0.73 \cdot T$ for any t. Thus

$$\mu_t = \mu_0 \exp\Big(-\eta \cdot \sum_{t' \leq t} (\widetilde{\rho} - p_{t'}(b_{t'}))\Big) \leq \mu_0 \exp\Big(\eta \cdot 0.73 \cdot T\Big) < \mu_0 \cdot (3\mu_0) = 3\mu_0^2,$$

where the last inequality follows from $\mu_0 \cdot \exp\Big(-\eta \cdot 0.73 \cdot T\Big) > 1/3$. Consequently, we have $b_t = \frac{1}{\mu_t} \cdot \frac{\lambda_t + 1}{\lambda_t} \geq \frac{1}{3\mu_0^2}$ and thus spend at least $\frac{1}{72\mu_0^4}$ in all iterations before the budget is depleted. It is straightforward to see that the maximum possible total value under this condition is obtained when bidding exactly $\frac{1}{3\mu_0^2}$ (and spending exactly $\frac{1}{72\mu_0^4}$) per iteration until the budget depletes (Observation D.1). This gives a value of $\frac{1}{12\mu_0^2}$ per iteration, and the budget is depleted after $\widetilde{\rho} \cdot T / \Big(\frac{1}{72\mu_0^4}\Big) = \Big(\frac{T}{200\mu_0^4}\Big) / \Big(\frac{1}{72\mu_0^4}\Big) = \frac{72 \cdot T}{200}$ iterations. The total value obtained by sequential pacing in this case is at most $\frac{3T}{100\mu_0^2}$, which is at least $\Omega(T)$ smaller than the optimal value of $\frac{T}{20\mu_0^2}$ (as μ_0 is $O(1)$ by assumption).

This completes our argument that given any instantiation of the sequential pacing algorithm, there exists an instance, which satisfies all the assumptions we need for the min pacing algorithm, such that the sequential pacing algorithm either incurs at least $\Omega(T)$ violation of the ROS constraint, or has a regret at least $\Omega(T)$.

E. Supplementary Material for Empirical Study

E.1. Semi-synthetic Dataset Construction

As we mentioned in Section 4, for confidentiality and advertiser privacy reasons, we evaluate the three pacing algorithms on a semi-synthetic dataset based on actual online advertising auctions. In particular, we focus on advertising campaigns from an online advertising platform that use a bidding product which is captured by our optimization formulation (1). More specifically, an advertiser bids (and therefore also pays) for clicks, i.e., submits bids for cost-per-click, and the objective is to maximize expected acquisitions (e.g. site visits, calls, conversions) with constraints on total spend being below an input budget and average cost per acquisition below an input target cost ($tcpa$). In our formulation (1), this corresponds to:

1. The value v_t is equal to $tcpa \cdot pconv_t$, where $pconv_t$ is the probability of a conversion conditioned on a click (note both $tcpa$ and $pconv_t$ are taken to be independent of the bid; while it is obvious for $tcpa$ to be independent of the bid, $pconv$'s independence is supported by empirical studies (Varian, 2009));
2. The allocation $x_t(b_t)$ is the number of clicks won by the advertiser at a bid of b_t;
3. The payment $p_t(b_t)$ is the cost of the clicks won at a bid of b_t.

Since we study the stochastic setting where the functions $x_t(\cdot), p_t(\cdot)$ are drawn i.i.d. from some distribution, our dataset consists of a set of generative models. The parameters of the generative model for any given (actual) advertising campaign we study are derived from the performance of that campaign in the (actual) auction.

Generative Model. We will use a i.i.d. stochastic model to generate the $x_t(b_t)$ and $p_t(b_t)$ in iteration t as a function of bid b_t (as discussed earlier, we slightly abuse notation to use b_t to be the multiplier to $tcpa \cdot pconv$). We use a Poisson distribution for $x_t(b_t)$ (i.e. number of clicks in an iteration at a bid b_t). With parameter λ, its probability mass function is $f(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$. The parameter λ is the expected number of clicks in an iteration, and we set it using the bidding landscape. In particular, for the model corresponding to a campaign C, the expected number of clicks at bid b_t in an iteration would be $\lambda_C(b_t) = click_C(b_t)/T$ where T is the total number of iterations in a day. In our empirical evaluation, we pick $T = 144$, which translates to each iteration being a 10-minute period, i.e., the dual variables of the algorithms are updated every 10 minutes instead of after every auction. We also derive from the bidding landscape a cost-per-click $cpc_C(b_t) = \frac{cost_C(b_t)}{click_C(b_t)}$. Both $cost_C(\cdot)$ and $click_C(\cdot)$ are model parameters derived from the bidding landscape which we discuss next.

To summarize, the value, click and cost at bid b_t are as follows:

$$v_t = tcpa(C) \cdot pconv_t(C), \qquad x_t(b_t) \sim Poisson(\lambda_C(b_t)), \qquad p_t(b_t) = x_t(b_t) \cdot cpc(b_t) \cdot noise_p$$

where we introduce i.i.d. non-negative multiplicative noise $noise_p$ with expected value 1 to the cost. In our evaluation, we use a Gaussian distribution centered at 1 with standard deviation 0.1 and truncated to be within $[0, 2]$ (so it's non-negative and has expected value 1). Also, when empirically evaluating the tCPA campaigns, the conversion rates $pconv_t(C)$ are drawn from a Gaussian distribution centered at the average $pconv$ (derived from the bidding landscape) of the campaign with a standard deviation of 0.1 and truncated to be in $[0, 2]$.

Bidding Landscape. To see how the auction performance of a campaign determines its model parameters in the generative model, it is useful to begin with the notion of a bidding landscape. For each campaign C, we construct a bidding landscape as a function from bids to the (predicted) number of clicks and cost. This is done first at a per-query level using auction simulation. In more detail, for an ad opportunity (a.k.a. query) q where campaign C is eligible to show its advertisement, we look at the logged bids of all the other campaigns participating in the auction for this query q, and simulate the auction for any bid b of C to know if/where C's ad would be shown. This gives us the predicted number of clicks and cost per click corresponding to any particular bid b, and we refer to them as $click_{C,q}(b)$ and $cost_{C,q}(b)$. In our model, we use the actual (i.e., advertiser submitted) target cost per acquisition of C as $tcpa(C)$, and the logged average predicted conversion probability generated by the production machine-learning model as $pconv_q(C)$. For a query q, bids are given by $b = k \cdot v_q$ where $v_q = tcpa(C) \cdot pconv_q(C)$ is the value of the query.

We aggregate these single-query landscape functions to get C's daily bidding landscape by summing up the respective functions over all the queries in a day, e.g., $click_C(k) = \sum_q click_{C,q}(k \cdot v_q)$, and $cost_C(k) = \sum_q cost_{C,q}(k \cdot v_q)$. Note that these functions are non-decreasing in k. The per-query bidding landscapes are inherently step functions represented by the various bid thresholds that makes C's ad to be displayed at various positions (or not displayed at all). While the aggregated landscapes are already smoother than the per-query landscapes, we further smooth the aggregated landscapes by linearly interpolating between consecutive thresholds. See Figure 7 for an example of the aggregated daily bidding landscape of an ad campaign.[4]

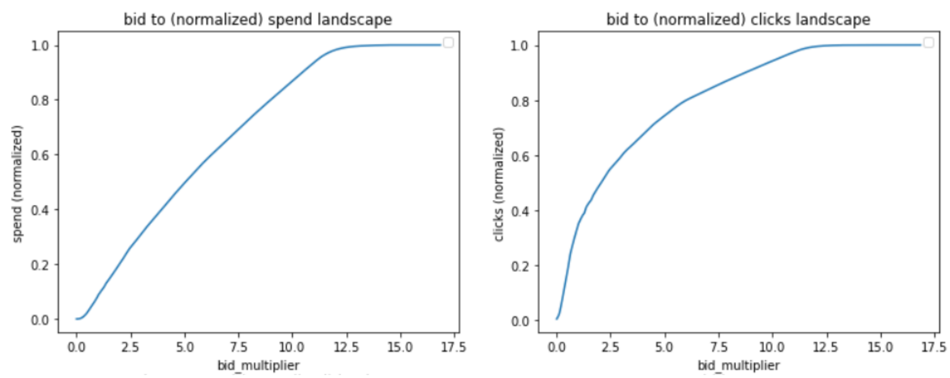


Figure 7. The bidding landscape of an example campaign. The x-axis is the bid (as a multiplier to value), and the y-axis are the daily cost and number of clicks (all normalized to be in $[0, 1]$) respectively.

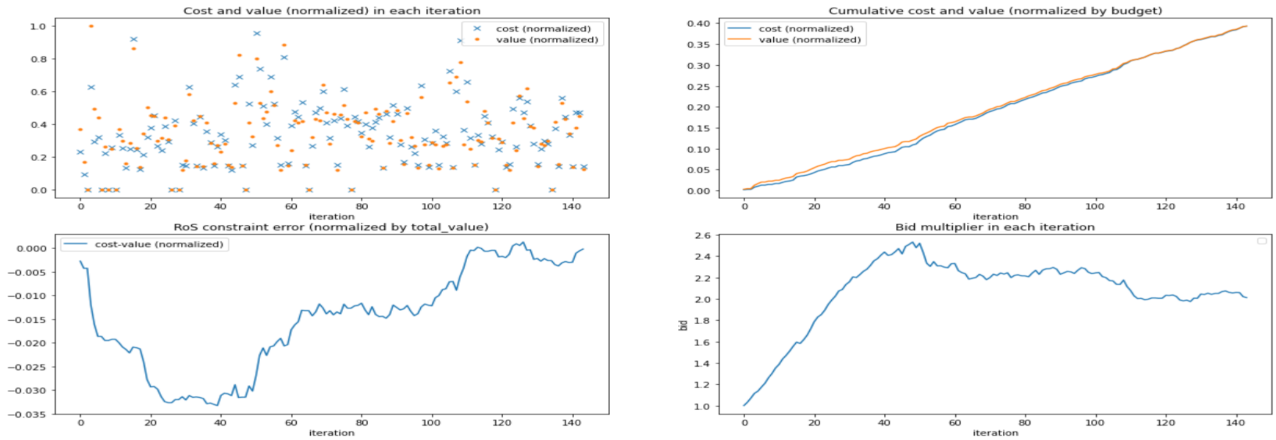
E.2. Empirical Evaluation

In our empirical study, our dataset includes 10^5 randomly selected campaigns, and for each campaign, we set the budget constraint (i.e. ρT in (1)) using its actual daily budget B.

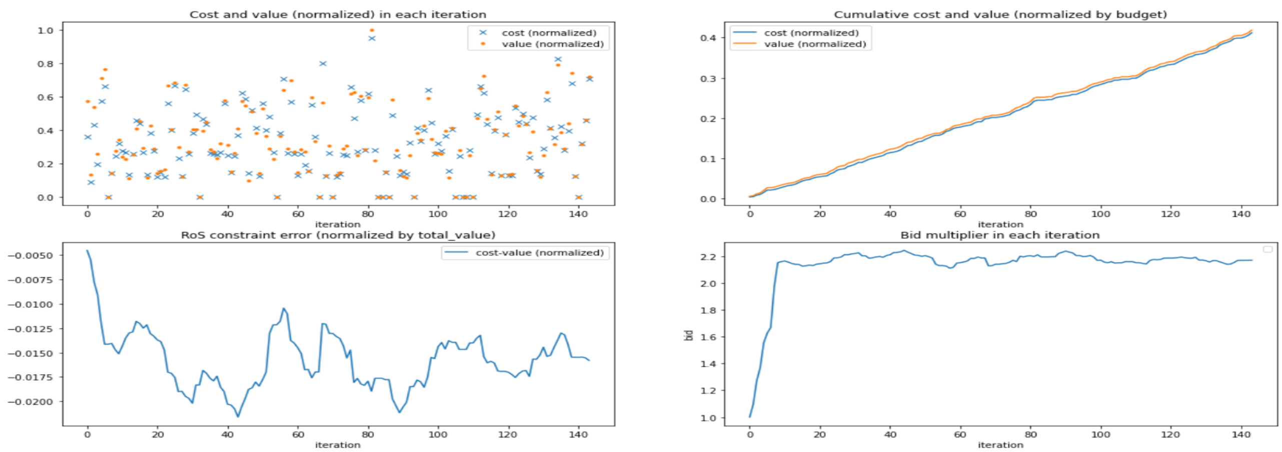
We divide the day into 10-minute periods and use $T = 144$, and simulate an algorithm on a particular campaign as follows. In each iteration, after the algorithm gives the bid it wants to submit, we compute the allocation and payment x_t, p_t using our generative model to get the number of clicks and cost of that iteration, and let the algorithm update the bid for the next iteration. We sum up the total cost and value through all T iterations. For the budget constraint, we follow the common practice to always strictly enforce it as follows: if in an iteration the generated cost is larger than the remaining budget, we modify that iteration's cost and value both to be 0. We do not enforce the ROS constraint strictly[5], but of course, measuring

[4]We normalize the values of click, value and cost in all the plots of this section, so the quantities shown do not represent real traffic or revenue.

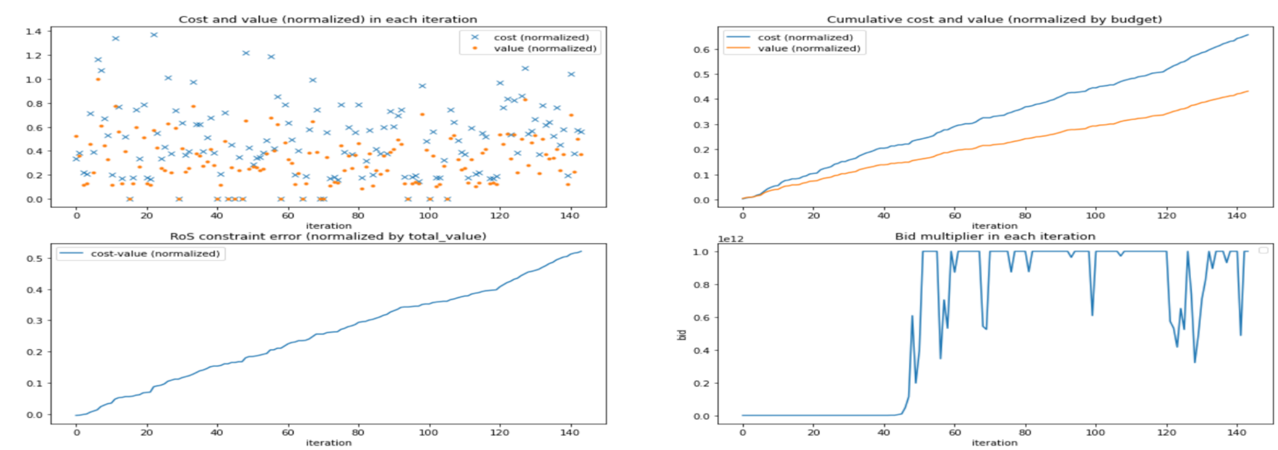
[5]Note that it is always possible for an ROS constraint to be temporarily violated after t rounds, but in the $t + 1$-th round it could become satisfied because of a really high value query coming through at low cost. Therefore it is suboptimal to stop serving right after ROS constraint gets violated in a round. This is not the case for budget constraint: once violated, it always remains violated because cumulative spend is monotonically increasing.



(a) Dual-optimal Pacing Algorithm



(b) Min Pacing Algorithm



(c) Sequential Pacing Algorithm

Figure 8. Simulation of the dual-optimal bidding (top), min bidding (middle) and sequential bidding algorithms (bottom) on an example campaign. We plot the per-iteration $value$ and $cost$ (normalized so $value \in [0, 1]$), cumulative $value$ and $cost$ (normalized by budget), cumulative ROS error (as $cost - value$ normalized by total $value$), and bids k.

how much the different algorithms violate the ROS constraint is an important aspect of this study and will be discussed here. In Figure 8 we visualize the pacing algorithms on an example campaign.

For each campaign, we simulate an algorithm 10 times to take the average (total) $spend$ and $conv\_val$ as the result of the algorithm on that campaign. For each algorithm, we take the 10^5 pairs of $(spend, conv\_val)$ from all the campaigns, and arrange them into buckets based on the relative ROS constraint error[6] $\max(0, spend/conv\_val - 1)$. For each bucket, we sum up the $conv\_val$ of all the campaigns in it. Moreover, for each algorithm, we do a grid search over the step-sizes used in the dual variables' updates. Each pair of step-sizes (one for each dual variable) is evaluated over the entire dataset, and for each algorithm we pick the best pair of step-sizes according to the total $conv\_val$ in the bucket of zero ROS constraint error. We compare the results associated with the best step-sizes for each algorithm.

E.3. Benchmark

For each campaign, our benchmark (10) (included again below) is the fluid relaxation of (1), but restricted to uniform bidding, i.e., $b_t = k \cdot v_t$ for all t.

$$\begin{array}{ll} \underset{k\geq 0}{\text{maximize}} & \sum_{t=1}^{T} \mathrm{E}[v_t \cdot x_t(k\cdot v_t)] \\ \text{subject to} & \sum_{t=1}^{T} \mathrm{E}[p_t(k\cdot v_t)] \leq \sum_{t=1}^{T} \mathrm{E}[v_t x_t(k\cdot v_t)], \\ & \sum_{t=1}^{T} \mathrm{E}[p_t(k\cdot v_t)] \leq \rho T\,. \end{array}$$

It is easy to see that in the stochastic i.i.d. model, the optimal value of (10) is an upper bound on the expectation of the ex-post optimal value. In our generative model, by design we have

$$conv\_val_C(k) = \mathrm{E}[v_t x_t(b_t)] = \frac{tcpa(C)}{T} \cdot \mathrm{E}[pconv_t(C) \cdot click_C\,(k \cdot tcpa(C) \cdot pconv_t(C))]\,,$$
$$spend_C(k) = \mathrm{E}[p_t(b_t)] = \mathrm{E}\left[x_t(b_t)\frac{cost_C(b_t)}{click_C(b_t)}\right]\mathrm{E}[noise_p] = \frac{1}{T}\,\mathrm{E}[cost_C\,(k \cdot tcpa(C) \cdot pconv_t(C))]\,,$$

where the expectation is taken with respect to the distribution of conversion probabilities of the different queries.

Our benchmark for campaign C in (10) becomes

$$\begin{array}{ll} \underset{k\geq 0}{\text{maximize}} & conv\_val_C(k) \\ \text{subject to} & spend_C(k) \leq conv\_val_C(k), \\ & spend_C(k) \leq \rho\,. \end{array} \tag{16}$$

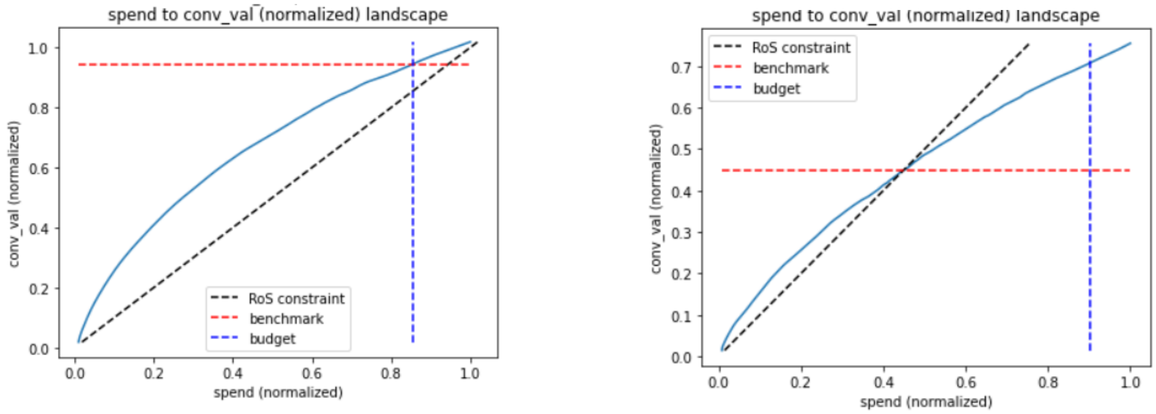
In our experiments, we approximate $conv\_val_C(k)$ and $spend_C(k)$ by performing a certainty equivalent approximation in which we replace random quantities (i.e., the predicted conversion probabilities) by their expected values. We solve the above optimization problem on the bidding landscape functions by finding the largest bid multiplier k^* such that $spend_C(k^*)$ is below C's budget and $conv\_val_C(k^*) \geq spend_C(k^*)$. Such multiplier k^* is easy to find using a line search since our landscape functions are all monotone in k. Furthermore, the restriction to uniform bidding in (10) is without loss of generality when the $conv\_val_C(k)$ versus $spend_C(k)$ function is concave, which qualitatively holds in our data (e.g. Figure 9).

We use $conv\_val_C(k^*)$ as computed above as the benchmark for C (see Figure 9 for examples). Note this captures the expected optimal solution, but algorithms running on the generative model of C may achieve better ex-post value than the benchmark due to the stochasticity of the model. We add up the expected optimal value over all campaigns as the overall benchmark. Figure 9 shows the pairs of spend and conversion value levels that can be achieved by varying the bidding multiplier k for a typical campaign. The achievable curves $(spend_C(k), conv\_val_C(k))_{k\geq 0}$ lie in $\mathbb{R}^2_+$, start at the origin for $k = 0$, increase along both axis as the bid multiplier increases, and end at $k \to \infty$.

E.4. Stability and Convergence

We observe that the trajectory of bidding multipliers generated by the dual-optimal and min pacing algorithms converge to the optimal solution of the benchmark (10). Figure 8 shows a representative campaign for which the ROS constraint is

[6]ROS constraint states that $spend \leq conv\_val$. So a constraint violation would imply $spend > conv\_val$, i.e., $spend/conv\_val - 1 > 0$.



(a) Budget constraint is binding.

(b) ROS constraint is binding.

Figure 9. The optimal operating points of an example campaign. The achievable curve (solid blue) delineates the pairs of spend-conversion value pairs that can be achieved by different bidding multipliers. The black diagonal dotted line captures the ROS constraint (feasible pairs should lie above this line), the blue vertical dotted line captures the budget constraint (feasible pairs should lie to the left of this line). The optimal operating point is the smallest of the intersection points of the achievable curve with one of the constraints and is shown using the red horizontal dotted line. In (a) the budget constraint is binding, while in (b) the ROS constraint is binding.

binding in the benchmark (but the budget constraint is not). After a small learning phase, the dual-optimal pacing algorithm converges to the optimal multiplier of around $k^* \approx 1$. The return-on-spend constraint is mostly obeyed and the total spend is smaller than the budget.

For the sequential pacing algorithm, however, we do not observe the convergence of bid multipliers. In Figure 8, it can be seen that the bid multipliers generated by the sequential pacing algorithm for the same campaign are highly unstable. Moreover, the ROS constraint is violated by a significant amount and the budget is exactly depleted by the end of the horizon. Interestingly, the behavior of the sequential pacing algorithm is driven by conflicting feedback loops between the budget and ROS pacing services. Recall that, at optimality, only the ROS constraint should bind. Initially, as the ROS pacing service detects a violation of the ROS constraint, it starts increasing its dual variable λ_t to satisfy the constraint. This results in a smaller bid multiplier k_t and reduced spend. The budget pacing service, however, believing that the budget constraint is not binding reacts to the lower spend by decreasing its dual variable μ_t, which in turn, results in a higher multiplier. These two opposing feedback loops generate unstable dynamics and one constraint ends up being violated. Similar behaviors are observed across campaigns even when the budget constraint is binding.