Stochastic bandits for multi-platform budget optimization in online advertising

V. Avadhanula
Facebook
Menlo Park, CA, US
vas1089@fb.com

R. Colini-Baldeschi
Facebook
London, UK
rickuz@fb.com

S. Leonardi
Sapienza University of Rome
Rome, Italy & Facebook
London, UK
leonardi@diag.uniroma1.it

K. A. Sankararaman
Facebook
Menlo Park, CA, US
karthikabinavs@fb.com

O. Schrijvers
Facebook
Menlo Park, CA, US
okke@fb.com

ABSTRACT

We study the problem of an online advertising system that wants to optimally spend an advertiser's given budget for a campaign across multiple platforms, without knowing the value for showing an ad to the users on those platforms. We model this challenging practical application as a Stochastic Bandits with Knapsacks problem over T rounds of bidding with the set of arms given by the set of distinct bidding m-tuples, where m is the number of platforms. We modify the algorithm proposed in Badanidiyuru *et al.,* [11] to extend it to the case of multiple platforms to obtain an algorithm for both the discrete and continuous bid-spaces. Namely, for discrete bid spaces we give an algorithm with regret $O\left(OPT\sqrt{\frac{mn}{B}} + \sqrt{mnOPT}\right)$, where OPT is the performance of the optimal algorithm that knows the distributions. For continuous bid spaces the regret of our algorithm is $\tilde{O}\left(m^{1/3} \cdot \min\left\{B^{2/3}, (mT)^{2/3}\right\}\right)$. When restricted to this special-case, this bound improves over Sankararaman and Slivkins [34] in the regime $OPT << T$, as is the case in the particular application at hand. Second, we show an $\Omega\left(\sqrt{mOPT}\right)$ lower bound for the discrete case and an $\Omega\left(m^{1/3}B^{2/3}\right)$ lower bound for the continuous setting, almost matching the upper bounds. Finally, we use a real-world data set from a large internet online advertising company with multiple ad platforms and show that our algorithms outperform common benchmarks and satisfy the required properties warranted in the real-world application.

KEYWORDS

Online Advertising, Multi-armed Bandits

1 INTRODUCTION

As online advertising has proliferated, ad campaigns have moved from ad-hoc bidding for individual users, to campaigns that try to reach massive audiences while respecting marketing budgets. In recent years this has led online advertising marketplaces to offer budget management solutions that bid on advertiser's behalf to optimally spend a given budget. But, as the online advertising ecosystem grows in size and complexity, those budget management solutions have to take an increasing number of factors into account when optimizing an advertiser's spend.

One such problem that is becoming increasingly ubiquitous in online advertising is designing a bidding strategy to spend an advertiser's budget optimally *across a set of different products*, where each product may have a different user base, competition, and advertising dynamics. Large tech companies have platforms that allow the advertisers to set-up campaigns that can be delivered to users across different internal products (in this paper we will use the unified terminology of *platforms*). The key challenge this leads to is that a campaign with a single budget has to acquire impressions from a variety of platforms that provide different value possibly due to a different user-base and different prices owing to different competition. Further challenges arise from the reality that advertisers often do not have a good understanding of their valuation of showing an ad to users spread across different platforms. Consider, for example, the case of an eCommerce business that advertises a specific good on their website. While they may have a good understanding of how much a conversion—i.e. the user purchases the good—is worth, they may have to pay for a click without knowing how likely the user is going to convert given a click on each of these platforms. Therefore, there is a need for developing budget management solutions that can optimize campaign delivery across multiple platforms in face of this uncertainty and budget constraints.

In this paper, we provide a rigorous mathematical model to study the aforementioned application and provide algorithms that are

optimal. We also expect these algorithms to be extremely fast and thus, can be deployed in real-world bidding engines where the system has only a few milliseconds to set bids. First, we describe some practical constraints that the bidding engines face before describing the various attempts at modeling this problem. In modern bidding engines, a bid is composed of multiple components. Most components of this bid is set for a *batch* of requests, before adding a platform specific multiplier that depends on the request. Thus, from the perspective of the central bidding engine, the goal is to decide bids for each of the platforms and this bid will be applied to the next batch of requests in the auction stage. After setting this main bid, the engine only sees a feedback of the total value received from each platform and the total budget that is consumed.

Motivated by the aforementioned example, in this work, we consider the *multi-platform placement optimization with unknown valuations*. More specifically, we study the problem of an advertiser that wants to optimally spend a given budget B over a time horizon T across m platforms. Those platforms select the winners via second price auctions. The goal is to design a policy that submits, at each time t, m bids $b_t(1), ..., b_t(m)$, one for each of the m-platforms. The bids are chosen such that the advertiser maximizes the cumulative reward over all the platforms while ensuring that the aggregate payments across all m platforms and the time horizon T do not exceed the allocated budget B. In other words, the objective of the advertiser is to learn the valuation of bidding on each of those platforms while simultaneously maximizing their cumulative rewards.

The academic community has worked actively to provide methods for doing this effectively when a campaign delivers only in a single platform. The studied methods focused on solving the problem either by limiting the set of auctions that bidders are eligible for, e.g. [1, 9, 22, 26, 30], or by adjusting the bids that enter the auction, e.g. [13, 16, 17, 19, 23, 33]. The latter are of particular interest, because they can either be implemented by the online advertising platform (as is common), but also by an advertiser themselves. However, the work on budget-management for online advertising has so far not focused on two important concerns: firstly, the marketing budget should be used over different platforms where users on different platforms are valued differently and platforms have different levels of competition, and secondly, advertisers may not have a clear understanding of their valuation for showing an ad to different users across these platforms.

The problem of learning to bid without knowing your value naturally leads to a *multi-armed bandit* formulation of the problem, and for a single platform has been studied without a budget constraint by Feng et al. [20], and with a budget constraint by [6, 21, 36], though none of these approaches consider bidding on multiple platforms. In this work, we formulate the problem as a stochastic bandits with knapsacks problem (BwK), introduced in the seminal work by Badanidiyuru et al. [10] and generalize the approach to the multi-platform optimization problem ([34]). One of the main contributions of the present paper is in highlighting the salient features of BwK that need to be adapted to design a computationally tractable algorithm that can compute the optimal bid vector from a combinatorial feasible bid set and ensuring that the performance (as measured by regret) of our algorithm scales optimally in both the number of platforms and the optimal value in

hindsight. [34] consider a more general problem where the combinatorial feasible set can be any general matroid. Thus, their algorithm is much more complicated and their bounds in the regime when $OPT << T$ is weaker than the one in the present paper. In this work, the feasible bid set is the special case of partition matroid which our algorithm specifically exploits and thus, does not require the rounding procedure required in [34].

Before we present our modeling of this problem with an exponential set of bidding m-tuples, we like to discuss some alternative formulations and their drawbacks. A first natural attempt at modeling this problem is to set it up as a *multi-armed bandit* problem where we have $n \times m$ arms with a group of n arms denoting the set of discrete bids that can be chosen for each of the m platforms. The game proceeds in $T \times m$ rounds, where T corresponds to the total number of steps for which the bidding engine has to decide the main bid. We split the $T \times m$ rounds into T phases where within each phase the algorithm plays for m rounds and sets the bid for each of the m platforms sequentially. At the end of this phase, the algorithm receives a feedback, *i.e.,* the total value and the budget consumed on each platform during this phase. The goal of the algorithm is to maximize the total value while respecting the total budget constraint. We argue that this model has two major drawbacks and thus, is not suitable for this application. First, at any given time-step $(t, i) \in ([T], [m])$, the algorithm cannot choose all the arms, but only a subset of the arms (*i.e.,* those that correspond to platform i). In a typical multi-armed bandit problem, we assume that all arms are available at all time-steps. We could model it as the more general *sleeping bandits* [27] problem but note that even without budget constraints this problem requires a change in benchmark (*i.e.,* the best ordering of arms) which does not translate to a meaningful notion in our application. Second, in MAB problems, we assume that the feedback received by the algorithm is immediate. However, in this modeling approach we would only receive the feedback at the end of the phase. Algorithms for *delayed feedback* models compare against *static* policies, while with budget constraints it is known that (*e.g.,* [10]) optimal dynamic policies can obtain at least twice more reward than the optimal static policy.

The second natural attempt to model this problem is to consider the *contextual* bandit framework. We have n arms (corresponding to each of the discrete bids) and T time-steps. At each time-step, the algorithm first sees a context x_t (in this case it corresponds to the platform on which the bids need to be chosen on) and then chose an arm that is a function of history and context. After every m rounds, the algorithm receives the feedback for the last m rounds (note even the bidding engine sets the bid for a platform, the feedback is collected over the batch and given back to the engine at the end of the phase). This modeling also has a two major drawbacks. First, as before, we need to handle delayed rewards (now in the much harder contextual bandit setup) for which to obtain algorithms with provable guarantees we need to make further assumptions (that may not hold in practice) such as linear rewards, static optimal policy. Second, contextual bandit problems with budget constraints are significantly harder (*e.g.,* [4]) where the regret bounds require the *large* budget assumption. In the context of online advertising for any campaign the total budget is very small compared to the number of auctions it participates in and thus, such large budget assumption is not reasonable.

Owing to the above difficulties, we model this problem as a variant of the *Bandits with Knapsacks* problem with combinatorial actions. We assume that we have $n \times m$ arms, where a group of n arms correspond to the possible bids for each of the m platforms. At each time-step, the algorithm chooses one bid per platform (a total of m arms) and sets this to be the bid for the next phase of auctions. At the end of this phase, the algorithm receives the feedback (the value and the budget consumed) and the algorithm can adapt its bid for the next phase. If for any platform, the bidding engine chooses not to bid for this phase, the algorithm simply chooses the 0-bid which ensures that it will not win any auction. Moreover, in the generalization later we show that the bids need not be discrete and can be a continuous value in the range $[0, r]$ for some known value $r \geq 1$. This modeling removes the difficulties in the above natural approaches: the benchmark for an optimal dynamic policy is indeed a best possible algorithm the bidding engine can employ, the feedback for each of the arms is obtained immediately at the end of the phase and we do not need to make assumptions on the ratio of budget to the total number of time-steps. *Apriori*, a new challenge this modeling approach introduces is that the algorithm needs to choose from an exponentially available set of choices at each time-step. However, as [34] show, both the algorithm and the regret bounds can overcome this and only depend on a polynomial function of the total number of platforms.

1.1 Overview of Contributions

We present algorithms for budget management in online advertising across multiple bidding platforms. The paper presents five main contributions:

- **Tractable mathematical model and algorithms.** The first contribution of this paper is to identify the correct mathematical model to study this problem. As mentioned earlier a number of natural choices fail since they lead to difficulties in obtaining an algorithm with provable guarantees. We study this problem in the framework of Stochastic bandits with knapsacks (BwK) introduced by Badanidiyuru et al. [10]. The goal is to design online bidding strategies that use the information available from the past rounds of ad auction to achieve suboptimal regret with respect to the optimal stochastic strategy that knows in advance the distributions of price and valuation on each platform. Unlike [34] we do not require the algorithm to invoke the rounding sub-routine at each time-step. The rounding sub-routine runs in $O((mn)^2)$ time-complexity per time-step and cannot be amortized across rounds. Thus, this improves the running time of our algorithm when the bid-space and the number of platforms become very large. On the other hand, similar to [34], our algorithm requires us to solve a linear program at each time-step. In practice, this can be fast, since we can cold-start the solution for each time-step with the solution of the previous time-step and thus, the computations can be amortized across the T time-steps to an average of a constant number of steps per-round.
- **Algorithm and regret analysis**. In Section 3, we present a bandit algorithm that achieves a regret bound $O\left(OPT\sqrt{\frac{mn}{B}} + \sqrt{mnOPT}\right)$ for discrete bid spaces, (where m

is the number of platforms, n the size of the discrete bid space, B the budget and OPT is the performance of the optimal algorithm that knows the distributions). This improves over the regret bound presented in [34] from $O(\sqrt{T})$ to $O(\sqrt{OPT})$ while keeping the dependence on the other parameters same. Our approach follows very closely the Primal Dual algorithm with knapsack constraints of [10]. However, we also show how to reduce the problem of finding the optimal bid vector to the problem of maximizing the ratio between a linear function of the UCB of the valuations and a linear function of the LCB of the costs, and this problem can be solved in polynomial time for a set of totally unimodular linear constraints [8]. In Section 4, we use the discretization idea introduced in [11], to discretize the continuous bid space in $[p_0, 1]$, with p_0 being the minimum reserve price across all platforms, by using an ϵ grid. The regret bound $\tilde{O}\left(\frac{(mv_0)^{1/3}}{p_0^{2/3}} \cdot \min\left\{\left(\frac{Bv_0}{p_0}\right)^{2/3}, (mT)^{2/3}\right\}\right)$ is obtained by adding the discretization error $\frac{B\epsilon v_0}{p_0^2}$, with v_0 being the maximum expected valuation across all platforms, to the regret for the discrete setting with $n = 1/\epsilon$ bid values. For the special case of single platform we improve over the $O(T^{2/3})$ bounds in [37] for the more challenging and realistic setting of $B = o(T)$. Additionally, the algorithms in [37] require the information on whether the algorithm won the auction or not at each time-step; while our algorithms do not need this information. Note, that ours is a more realistic assumption, in general, since advertising systems have *checks and balances* in place, after an auction phase and thus, impression/conversion information does not directly translate to whether an ad won an auction or not.
- **Lower bounds**. In Section 5 we complement our algorithmic results with a $\Omega\left(\sqrt{mOPT}\right)$ lower bound for the discrete case and an $\Omega\left(m^{1/3}B^{2/3}\right)$ lower bound for the continuous case thus showing that our algorithmic results are close to optimum despite the complexity of our setting.
- **Experiments on real data**. In Section 6 we evaluate our online bidding strategies on real-world dataset with multiple platforms and several budget limited advertisers obtained from the logs of a large internet advertising company. We compare our algorithms against the total reward of the optimal LP value computed on the mean valuation and prices and on two different baselines. The first baseline is the naive UCB algorithm that ignores the budget constraints and maximizes the rewards. The second algorithm is the *LuekerLearn algorithm* from [37] adapted to multiple platforms and stochastic valuations. Our experimental results show that we outperform both the benchmarks in efficiency (total regret accumulated). We show how the various algorithms compare as a function of budget and number of platforms. Moreover, we also observe that our proposed algorithm uniformly depletes the budget and always runs till the specified time-limit, while both the baselines exhaust budget very early. Uniform budget spend is an *expectation* many advertisers have and

thus, this behavior of our algorithm is extremely desirable for practical deployment.

1.2 Additional Related Work

The problem of optimal bidding policies in second price auctions from the perspective of bidders/advertisers can be broadly categorized into four themes.

Budget management with known valuation. Early approaches to budget management with known valuations (and a single platform) have focused on shading an advertiser's value by a multiplier (often called the "pacing multiplier"). For example, the work of [19, 23, 33] implement bid shading strategies to maximize the advertisers ROI. What makes this approach appealing, is that the static problem (with known competitor bids) has an optimal solution with a *constant* pacing multipier, so the dynamic approach aims to find this constant. More recent work [13–17] have complemented this bid-manipulation literature with equilibrium analysis under the assumption that all advertisers use the same bid-shading approach.

An alternative to bid modification, is for a platform to restrict the number of auctions that an advertiser participates in. Work by Mehta et al. [30] give revenue guarantees for the online matching problem of users showing up over time. Subsequently bidder selection has been applied to multi-objective optimization for Generalized Second Price auctions in e.g. [1, 9, 22, 26]. While the work on bidder selection yields allocations that respect the budget constraint, advertisers are not best-responding to the competition and they can generally improve their allocation by lowering their bids.

Bidding with unknown valuations and no budget constraints. Feng et al. [20] focus on the problem of bidding on a platform without knowing the valuations for users. They model this as a stochastic bandit setting with partial feedback (they only learn the value if the ad won, and the action that the advertiser cares about was taken). They give an online learning algorithm with regret that grows logarithmically in the size of the action space. Their model considers more complex auction environments such as position auctions [18, 38], though it doesn't take into account multiple platforms or budget constraints.

Budget constrained bidding and unknown valuations. This line of literature combines the problem settings of the above two themes. More specifically, existing work consider the setting where the valuation as well as the distribution of the competing bids is unknown and focus on optimal bidding strategies that satisfy the specified budget, maximizes the cumulative rewards and learns the valuation of the item. To the best of our knowledge, the problem of learning the optimal bid in second price auctions was first considered by Amin et al. [6]. They focus on the discrete bids setting, where the feasible bids belong to a discrete set of bids and formulate the problem as Markov decision process with censored observations. They propose a learning algorithm based on the popular product-limit estimator and demonstrate empirically that their algorithm converges to optimal solution. Tran-Tranh et al. [36] followed up on the work of [6] and provided theoretical guarantees for the algorithm proposed in the later work. More specifically, they assume that the bids are sampled from a discrete distribution in $\{1, \cdots, C\}$ and establish $O(\sqrt{CB})$ regret bounds, where C is the number of

feasible bids and B is the total budget. The aforementioned regret bounds depend on some parameters of the competing bids distribution. Flajolet and Jailiet [21] consider a more general variant of the aforementioned problem by relaxing the assumption that the bids come from a discrete set and proposed an algorithm based on the probabilistic bisection (see [32]) and upper confidence bounds (UCB) family of algorithms. They establish $O(\sqrt{T})$ regret bound for the problem under the assumption that budget scales linearly with time, i.e $B = \Theta(T)$. All three works ([6], [36] and [21]) in this theme only focus on bidding strategies for a single platform and cannot be easily generalized to the multi-platform setting. Finally, in [31] it is studied the problem of joint bid/daily budget optimization of pay-per-click advertising campaigns over multiple channels. The authors formulate the problem with combinatorial semi-bandits, which requires solving a special case of the Multiple-Choice Knapsack problem every day, and it obtains an $O(\sqrt{T})$ regret bound. Differently from our work that performs on online budget optimization across multiple rounds, this last paper does static budget optimization in each single round.

Stochastic bandits with knapsack constraints. Stochastic bandits with knapsacks (BwK), introduced in the seminal work of Badanidiyuru et al. [10] is a general framework that considers the standard multi-armed bandit problem under the presence of additional budget/resource constraints in addition to the time horizon constraint. The BwK problem encapsulates a large number of constrained bandit problems that naturally arise in many application domains including dynamic pricing, auction bidding, routing and scheduling. [10, 11] presents two algorithms based respectively on balanced exploration and primal-dual also used in this work with performances close to the information theoretical optimum. More specifically, they establish a $O(\log(dT)(\sqrt{X\overline{OPT}} + \overline{OPT}\sqrt{m/B}))$, where X is the total number of arms, d is the number of constraints, B is the binding budget constraint and $\overline{OPT}$ is a "reasonable" upper bound on the cumulative optimal reward. A naive generalization of the BwK framework to our problem would result in a combinatorial number of arms (n^m possible bids). Recently, Sankarraman and Slivkins [34] extended the BwK framework to combinatorial action sets and established $O(\sqrt{XT})$ regret bounds, which is only optimal only when the cumulative reward is comparable to the length of the time horizon, which is not the case in many online auctions. In this work, we extend the BwK ([10]) framework to propose an optimal bidding strategy with performances guarantees that are near optimal even when the cumulative reward is much smaller in comparison to length of the time horizon.

The BwK framework [10] also has been extended with near-optimal regret bounds to contextual bandits (see [3, 5, 12]), settings with concave reward functions and convex constraints (see [2]). Adversarial bandits with knapsack are also extended in [24] to the combinatorial semi-bandit, contextual, and convex optimization settings. However, these problem settings are not immediately applicable to our problem.

2 AD PLATFORM OPTIMIZATION WITH UNKNOWN VALUATIONS

We model multi-platform real-time bidding for budgeted advertisers as a multi-armed bandit problem with knapsack constraints [11]. The advertiser has an available budget B for T rounds of the ad auction. At each time step $t \in T$, the bidder chooses a bid $\boldsymbol{b}_t \in [0, 1]^m$ where $b_t(i)$ is the bid for platform $i \in [m]$. We also consider the special case of discrete bid in section 3, each of the bid $b_t(j)$ is chosen from the finite set $\mathcal{B} = \{b_1, \dots, b_n\}$, where $b_j \in [0, 1], j \in [n]$. The set of arms $\mathcal{X} = \{\langle b(1), \dots, b(m)\rangle, b(i) \in \mathcal{B}, i \in [m]\}$ is therefore given by the set of distinct bidding m-tuples. Denote by $\boldsymbol{b} = \langle b(1), \dots, b(m)\rangle$ the generic bid vector and by $x_{\boldsymbol{b}} \in \mathcal{X}$ the corresponding arm.

At each phase t, each platform i runs many second price auctions among the bids that are received for this phase. When at time t, the bidder bids b_t across the platforms, bid $b_t(i)$ is entered on platform i. Let $p_t(i) \sim P(i)$ be the *critical bid*[1] for the bidder on platform i at time t (where $P(i)$ is a stationary distribution). If $b_t(i) \geq p_t(i)$ the bidder wins the auction (we break ties in favor of the advertiser), the price is equal to the critical bid $p_t(i)$ and at that point learns the realized value $v_t(i) \sim V(i)$ (where $V(i)$ is a stationary distribution). For ease of presentation, bids, critical bids and values are normalized so that $b(i), p(i), v(i) \in [0, 1]$ for all $i \in [m]$, and we'll use the terms critical bid and price interchangeably where appropriate [2].

We denote by

$$r_{x(\boldsymbol{b})} = \sum_{i \in [m]} v(i)\mathbb{I}[b(i) \geq p(i)] \qquad \text{and}$$

$$c_{x(\boldsymbol{b})} = \sum_{i \in [m]} p(i)\mathbb{I}[b(i) \geq p(i)]$$

the reward and the cost of arm $x_{\boldsymbol{b}}$ with critical bid vector $\boldsymbol{p}$ and reward vector $\boldsymbol{v}$. We denote by $\boldsymbol{b}_t = \langle b_t(1), \dots, b_t(m)\rangle$ the bid vector of the bidder at time t. We also use $\boldsymbol{p}_t = \langle p_t(1), \dots, p_t(m)\rangle$ for the critical bid vector that is realized at time t and $\boldsymbol{v}_t = \langle v_t(1), \dots, v_t(m)\rangle$ for the vector of rewards that is realized at time t.

In the stochastic bandit setting, critical bid $p(i)$ and reward values $v(i)$ are drawn at any time t, respectively, from the stationary independent distributions $P(i), V(i)$ for $i \in [m]$, unknown to the algorithm. We also denote with $\bar{r}_{x(\boldsymbol{b})} = E[r_{x(\boldsymbol{b})}]$ and $\bar{c}_{x(\boldsymbol{b})} = E[c_{x(\boldsymbol{b})}]$ the expected value of the reward and the cost of arm $x(\boldsymbol{b})$, where the expectation is taken over the critical bid $p(i) \sim P(i)$ and the reward value $v(i) \sim V(i)$.

The bidder must decide on the bidding vector $\boldsymbol{b}_t$ without the knowledge of the critical bid vector $\boldsymbol{p}_t$ and the reward vector $\boldsymbol{v}_t$. If $b_t(i) \geq p_t(i)$, the feedback provided to the advertiser upon bidding $b_t(i)$ is price $p_t(i)$ and utility $v_t(i)$. If $b_t(i) < p_t(i)$, the advertiser only learned that $b_t(i)$ is lower than the critical bid. The goal is to design an online bidding strategy that selects a bid vector $\boldsymbol{b}_t$ at each round t so that the time-averaged reward across the multiple platforms is maximized and the total cost paid by the algorithm does not exceed budget B.

---

[1]The critical bid is either the highest bid of the other bidders or the reserve price, whichever is higher.

[2]Note this is without loss of generality, since we can scale the values and multiply this scale in the regret bound.

Given that the advertiser is budget limited, we use random variable τ to indicate the first time the budget of the advertiser is exhausted, i.e.,

$$\tau = \min\left(T + 1, \min\left\{z \in [T] | \sum_{t=1}^{z} c_{x(\boldsymbol{b}_t)} > B\right\}\right)$$

We compare the reward obtained by the algorithm against the reward OPT of an optimal bidding strategy that knows the distributions $P(i), V(i)$ for $i \in [m]$, and it is allowed to submit a new bid vector at each time. Our goal is to bound the regret of the bidding strategy of the advertiser defined by

$$OPT - \mathbb{E}\left[\sum_{t=1}^{\tau} r_{x(\boldsymbol{b}_t)}\right],$$

where the expectation is taken over the random variables $\tau, p(i), v(i), i \in [m]$.

3 DISCRETE BID SPACES

We adapt the Primal Dual algorithm for bandits with knapsack constraints of [11]. We use two resources in our formulation. The budget B is consumed at rate equal to the price paid for each ad auction that is won by the advertiser. Furthermore, we have a second resource with budget B that consumes deterministically a budget B/T at each round of ad auction. The second resource ensures that the system can bid at most T times on each platform.

For the regret analysis of the algorithm, following the primal dual approach of [11], we compare with an upper bound on OPT given by the following LP formulation:

$$OPT_{LP} = \max \sum_{x \in \mathcal{X}} \xi_x \bar{r}_x \tag{1}$$

$$\sum_{x \in \mathcal{X}} \xi_x \bar{c}_x \leq B \tag{2}$$

$$\sum_{x \in \mathcal{X}} \frac{B}{T} \xi_x \leq B \tag{3}$$

$$\xi_x \geq 0, \forall x \in X \tag{4}$$

with ξ_x being the number of times arm $x \in \mathcal{X}$ is used during the T rounds.

According to the following claim, we can use the optimal LP solution in place of OPT for bounding the regret of the algorithm:

CLAIM 3.1. OPT_{LP} *is an upper bound on the value of the optimal dynamic policy:* $OPT_{LP} \geq OPT$.

In the algorithm, we denote by $\bar{v}_{ij}$ and $\bar{c}_{ij}$ the expected reward and the expected cost, respectively, obtained by bidding value b_j on platform i. We also denote by $v_{ij}^{UCB}(t)$ the upper confidence bound estimation of $\bar{v}_{ij}$ and by c_{ij}^{LCB} the lower confidence bound estimation of $\bar{c}_{ij}$. Concretely, they are defined as follows. Here $N_{ij}(t)$ denotes the number of times we bid i on platform j until

time t and $C_{rad} = \Theta(\log mnT)$.

$$v_{ij}^{UCB}(t) := \frac{1}{N_{ij}(t)} \sum_{t' < t} v_{ij}(t) \cdot \mathbb{I}[\text{bid } i \text{ on platform } j]$$

$$+ \sqrt{\frac{C_{rad} \cdot \bar{v}_{ij}}{N_{ij}(t)}} + \frac{C_{rad}}{N_{ij}(t)}$$

$$c_{ij}^{LCB}(t) := \frac{1}{N_{ij}(t)} \sum_{t' < t} c_{ij}(t) \cdot \mathbb{I}[\text{bid } i \text{ on platform } j]$$

$$- \sqrt{\frac{C_{rad} \cdot \bar{c}_{ij}}{N_{ij}(t)}} - \frac{C_{rad}}{N_{ij}(t)}$$

We denote by $\lambda_t(1)$ and $\lambda_t(2)$ the estimation of the dual variables computed by the algorithm. We have $d = 2$ in our case but we leave a generic term d for the purpose of an easy generalization to the case of an individual budget B_i for each $i \in [m]$.

Algorithm 1 Multi-platform bidding with parameter $\epsilon \in (0, 1)$

INITIALIZATION
Play arm x_{b_j} with bid $b_j = \{b(i) = b_j\}_{i \in [m]}, j \in n$ for an initial
UCB estimate of $\bar{r}_{ij}$ and an initial LCB estimate of $\bar{c}_{ij}$
$\lambda_1(1) = 1, \lambda_1(2) = 1$ is the estimate for η_1 and η_2
set $\epsilon = \sqrt{\ln(d)/B}$
for $t = n + 1, \ldots, \tau$ (i.e., until resource budget is exhausted) **do**
  Obtain the maximizer to the following

  $$\alpha_{i,j_i^*} =$$

  $$\arg\max_{\alpha_{ij} \in \{0,1\}} \quad \frac{\sum_{i \in [m]} \sum_{j \in [n]} r_{ij}^{UCB} \alpha_{ij}}{\lambda_t(1) \sum_{i \in [m]} \sum_{j \in [n]} c_{ij}^{LCB} \alpha_{ij} + \lambda_t(2)\frac{B}{T}}$$

  s.t.
  $$\sum_{j \in n} \alpha_{ij} = 1, \forall i \in [m]$$

  Play arm x_{b_t} with bid $b_t = \{b_{i,j_i^*}\}_{i \in [m]}$
  Update UCB estimate r_{ij}^{UCB} and LCB estimate c_{ij}^{LCB}
  Compute LCB estimate cost of arm x_t: $c_{1x_t}^{LCB} = \sum_{i \in m} c_{i,j_i^*}^{LCB}$,
  $c_{2x_t} = B/T$
  Update estimate of dual variables:
  $\lambda_{t+1}(1) = \lambda_t(1)(1 + \epsilon)^{c_{1x_t}^{LCB}}$
  $\lambda_{t+1}(2) = \lambda_t(1)(1 + \epsilon)^{c_{2x_t}}$
end for

In the initial exploration phase, bid b_i is submitted on all platforms at time $t = i$. The cost of the initial phase is therefore bounded by mn. At each round we compute the arm with bid vector that maximizes the *bang-per-buck* ratio between the Upper Confidence Bound of the reward and the Lower Confidence Bound of the normalized cost. The number of different arms that is exponential can be reduced in the analysis by pruning out suboptimal arms. The optimal arm according to the *UCB* and *LCB* approximations can actually be computed in polynomial time since this is the problem of optimizing a rational function subject to a set of linear constraints described by a totally unimodular matrix [8]. After the feedback is received, the UCB estimation of the rewards and the LCB estimation

of the costs are updated. Variables $\lambda(1), \lambda(2)$ are estimated using multiplicative weight update [29].

3.1 Analysis of the algorithm.

Once the problem of selecting the arm has been addressed through a separate optimization step, the analysis of the algorithm follows very closely the one of [11] with some care that allows to replace the exponential number of arms with mn in the regret bound.

Let $y_t(i) = \lambda_t(i)/\| \lambda_t \|_1, i = 1, \ldots, d$ be the normalized cost of the resources. For a parameter $\epsilon \in [0, 1]$, for every vector y, for any sequence of payoff vectors $c_1, \ldots, c_\tau \in [0, 1]^d$, Hedge's guarantee gives [29]:

$$\sum_{t=1}^{\tau} y_t^T c_t \geq (1 - \epsilon) \sum_{t=1}^{\tau} y^T c_t - \frac{\ln d}{\epsilon}. \tag{5}$$

In what follows, we denote by $c_{x_t} = \begin{bmatrix} c_{1x_t} \\ c_{2x_t} \end{bmatrix}$, the cost vector of arm x_t.

We must consider in the analysis the error that derives from using the UCB estimate for the rewards and LCB estimate for the costs. First of all, given that the maximum reward is mT, if we fail to have a clean execution, the loss is $O(mT)$. If we select $C_{rad} = \Theta(\log dTm)$, the probability of failure can be made much smaller than $1/(mT)$.

Let us denote by l_{x_t} the LCB estimate of c_{x_t} and by u_{x_t} the UCB estimate of r_{x_t}. Moreover, let $E_t = c_{x_t} - l_{x_t}$ be the error on the cost and $\delta_t = u_{x_t} - r_{x_t}$ the error on the reward. We also denote by $REW^{UCB} = \sum_t u_{x_t}^T \xi_t$ the UCB reward of the algorithm. The proofs of the following claim and theorem are given in the appendix.

CLAIM 3.2.

$$REW \geq OPT_{LP} \left[(1 - \epsilon) - \frac{mn+1}{B} - \frac{1}{B}\| \sum_{1 < t < \tau} E_t^T \xi_t \|_\infty - \frac{1}{B}\frac{\ln d}{\epsilon} \right]$$
$$- \left| \sum_{1 < t < \tau} \delta_t^T \xi_t \right|.$$

PROOF. The structure of this proof is similar to [11]. Let $\bar{y} = \frac{1}{REW^{UCB}} \sum_{n < t < \tau} (u_{x_t}^T \xi_t) y_t$ We prove the following inequalities:

$$
\begin{aligned}
B &\geq \bar{y}^T c \xi^* \\
&= \frac{1}{REW^{UCB}} \sum_{n < t < \tau} (u_{x_t}^T \xi_t)(y_t^T c \xi^*) \\
&\geq \frac{1}{REW^{UCB}} \sum_{n < t < \tau} (u_{x_t}^T \xi_t)(y_t^T l_{x_t} \xi^*) \\
&\geq \frac{1}{REW^{UCB}} \sum_{n < t < \tau} (u_{x_t}^T \xi^*)(y_t^T l_{x_t} \xi_t) \\
&\geq \frac{1}{REW^{UCB}} \sum_{n < t < \tau} (r^T \xi^*)(y_t^T l_{x_t} \xi_t) \\
&\geq \frac{OPT_{LP}}{REW^{UCB}} \left[(1 - \epsilon) y^T \left(\sum_{n < t < \tau} l_{x_t} \xi_t \right) - \frac{\ln d}{\epsilon} \right] \\
&\geq (1 - \epsilon) \frac{OPT_{LP}}{REW^{UCB}} \\
&\quad \left[y^T \left(\sum_{n < t < \tau} c_{x_t} \xi_t \right) - y^T \left(\sum_{n < t < \tau} E_t^T \xi_t \right) - \frac{\ln d}{\epsilon} \right] \\
&\geq \frac{OPT_{LP}}{REW^{UCB}} \\
&\quad \left[(1 - \epsilon)(B - mn - 1) - (1 - \epsilon) y^T \left(\sum_{n < t < \tau} E_t^T \xi_t \right) - \frac{\ln d}{\epsilon} \right],
\end{aligned}
$$

The first inequality follows from primal feasibility, the second inequality by the definition of $\bar{y}$, the third inequality by clean execution, the forth inequality by the rule of selection of the arm, the fifth

inequality follows from clean execution, and the sixth inequality from the guarantee of Hedge of equation 5.

For bounding the regret of the algorithm we finally use:

$$REW \geq REW^{UCB} - \sum_{n < t < \tau} (u_{x_t} - r_{x_t})^T \xi_t = REW^{UCB} - \left| \sum_{1 < t < \tau} \delta_t^T \xi_t \right|.$$

$\square$

We combine this claim with the following two bounds which can again be derived using the approach in [10].

$$\left| \sum_{n < t < \tau} \delta_t^T \xi_t \right| = O \left(\sqrt{C_{rad} mn REW} + C_{rad} mn \ln T \right),$$

$$\left\| \sum_{1 < t < \tau} E_t \xi_t \right\|_\infty = O \left(\sqrt{C_{rad} mn B} + C_{rad} mn \ln T \right).$$

By assuming $mn < B$ and $\epsilon = \sqrt{\frac{\ln d}{B}}$, we conclude with the following theorem:.

THEOREM 3.3. *The regret of the algorithm is bounded by*

$$O \left(OPT_{LP} \sqrt{\frac{mn}{B}} + \sqrt{mn OPT} \right).$$

PROOF. We use for the proof the following two claims on the UCB estimate of the reward and the LCB estimate of the cost are proved in [11]

CLAIM 3.4. $\left| \sum_{n < t < \tau} \delta_t^T \xi_t \right| = O \left(\sqrt{C_{rad} mn REW} + C_{rad} mn \ln T \right).$

CLAIM 3.5. $\left\| \sum_{1 < t < \tau} E_t \xi_t \right\|_\infty = O \left(\sqrt{C_{rad} mn B} + C_{rad} mn \ln T \right)$

We start from the claim of Claim 3.2:

$$REW \geq OPT_{LP} \left[(1 - \epsilon) - \frac{mn+1}{B} - \frac{1}{B} \| \sum_{1 < t < \tau} E_t \xi_t \|_\infty - \frac{\ln d}{\epsilon B} \right]$$
$$- \left| \sum_{1 < t < \tau} \delta_t^T \xi_t \right|$$

By assuming $mn < B/\ln dT$ and by choosing $\epsilon = \sqrt{\frac{\ln d}{B}}$, we bound the following three terms of the regret:

$$OPT_{LP} \left[\frac{mn+1}{B} + \frac{\ln d}{\epsilon B} \right] = O \left(OPT_{LP} \left[\frac{mn}{B} + \sqrt{\frac{\ln d}{B}} \right] \right);$$

$$\frac{OPT_{LP}}{B} \| \sum_{1 < t < \tau} E_t \xi_t \|_\infty$$
$$= O \left(\frac{OPT_{LP}}{B} \left[\sqrt{C_{rad} mn B} + C_{rad} mn \ln T \right] \right)$$
$$= O \left(\frac{OPT_{LP}}{\sqrt{B}} \sqrt{C_{rad} mn} \right)$$

and

$$\left| \sum_{1 < t < \tau} \delta_t^T \xi_t \right| = O \left(\sqrt{C_{rad} mn REW} + C_{rad} mn (\ln T) \right)$$

thus proving the claim. $\square$

4 CONTINUOUS BID SPACES

In the previous section we considered the discrete setting with n different bid values available on each platform for the advertiser. In this section we consider the continuous setting, with prices and valuations being real values in $[0, 1]$. Our approach will be to discretize the continuous bidding space to consider bid values that are multiple of some small value ϵ. The discretization of the bidding space will result into an additional regret loss. Most of the following analysis is therefore concerned with bounding the error of the discretization process.

Let the support of the critical bid distribution be in the interval $[p_0, 1]$, with p_0 being a small constant that can be considered as the reserve price for the ad auction. Let $r_b(i)$ and $c_b(i)$ the expected reward and the expected cost of bid b on platform $i \in [m]$, where the expectation is taken over the critical bid $p(i) \sim P(i)$ and the reward values $v(i) \sim V(i)$. We also denote by v_0 the maximum expected reward over all platforms if the auction is won.

The following Lemma shows that the discretization of the continuous bidding space is performed at the expense of a limited additive loss in the buck-per-bang ratio. The proofs are given in the appendix.

LEMMA 4.1. *For each bid $b \geq p_0$, and for each platform $i \in [m]$, it holds:*

(1) $c_{b+\epsilon}(i) \geq c_b(i)$
(2) $\frac{r_{b+\epsilon}(i)}{c_{b+\epsilon}(i)} - \frac{r_b(i)}{c_b(i)} \geq \frac{\epsilon v_0}{p_0^2}$

PROOF. The first part of the claim is straightforward since the expected cost can only increase with the value of the bid. We omit i for the proof of the second part of the claim.

Let $f(p)$ be the continuous density function of the cumulative distribution P. For the expected cost of bid b, it holds

$$c_b = \int_{p_o}^b p f(p) dp \geq r_b p_0 / v,$$

since the ratio between cost and reward of bid b is at least the minimum cost p_0 divided by the reward v that is obtained if the bid is accepted.

Denote $a = \int_b^{b+\epsilon} f(p) dp$. We obtain the following expressions for the cost and the reward at bid $b + \epsilon$:

$$r_{b+\epsilon} = r_b + v \int_b^{b+\epsilon} f(p) dp = r_b + a \cdot v$$

$$c_{b+\epsilon} = c_b + \int_b^{b+\epsilon} p f(p) dp \leq c_b + (b + \epsilon) a$$

We therefore have:

$$
\begin{aligned}
\frac{r_b}{c_b} - \frac{r_{b+\epsilon}}{c_{b+\epsilon}} &\leq \frac{r_b}{c_b} - \frac{r_b + a \cdot v}{c_b + (b+\epsilon)a} \\
&= \frac{r_b(b+\epsilon)a - ac_b v}{c_b(c_b + (b+\epsilon)a)} \\
&\leq \frac{r_b(b+\epsilon)a - ap_0 r_b}{r_b p_0 / v(p_0 + (b+\epsilon)a)} \\
&= \frac{a((b+\epsilon) - p_0)}{p_0 / v(p_0 + (b+\epsilon)a)} \\
&\leq \frac{av_0}{p_0^2},
\end{aligned}
$$

with the previous inequalities that follow from $c_b \geq r_b p_0 / v$, $c_b \geq p_0$, and $v \leq v_0$. By continuity of $f(b)$, $a = \int_b^{b+\epsilon} f(p)dp \leq \epsilon$ proves the result. □

The total loss in the regret due to the discretization error of the buck-per-bang ratio is therefore upper-bounded by $\frac{B\epsilon v_0}{p_0^2}$ to be added to the regret bound of Theorem 3.3. The set of distinct bid values is $b_j = j\epsilon, j \in [n]$, and therefore n can be replaced by $1/\epsilon$ in the claim of Theorem 3.3.

We therefore conclude with the following theorem:

THEOREM 4.2. *The regret of the algorithm with discretized bids is bounded by*

$$
\tilde{O}\left(\frac{(mv_0)^{1/3}}{p_0^{2/3}} \cdot \min\left\{\left(\frac{Bv_0}{p_0}\right)^{2/3}, (mT)^{2/3}\right\}\right)
$$

PROOF. By substituting $n = 1/\epsilon$ in the regret bound of the previous section, and by adding the additional loss in revenue given to the discretization, the bound on the regret is:

$$
\tilde{O}\left(OPT_{LP}\sqrt{\frac{m}{B\epsilon}} + \sqrt{\frac{mOPT}{\epsilon}} + \frac{B\epsilon v_0}{p_0^2}\right).
$$

Note that $OPT \leq OPT_{LP} \leq \min\left\{\frac{Bv_0}{p_0}, mT\right\}$. We have two cases. When $\min\left\{\frac{Bv_0}{p_0}, mT\right\} = \frac{Bv_0}{p_0}$ the regret is bounded by:

$$
\tilde{O}\left(\frac{v_0}{p_0}\sqrt{\frac{Bm}{\epsilon}} + \sqrt{\frac{mBv_0}{p_0\epsilon}} + \frac{B\epsilon v_0}{p_0^2}\right).
$$

Substituting $\epsilon = \frac{p_0^{2/3} \cdot m^{1/3}}{B^{1/3}}$ we obtain that the regret is upper-bounded by $\tilde{O}\left(\frac{m^{1/3}B^{2/3}v_0}{p_0^{4/3}}\right)$.

Likewise, when $\min\left\{\frac{B}{p_0}, mT\right\} = mT$, given that $mT < B$ if the problem is budget constrained, by setting $\epsilon = \frac{mp_0^{4/3}T^{2/3}}{Bv_0^{2/3}}$ we get the regret to be upper-bounded by $\tilde{O}\left(\frac{mT^{2/3}v_0^{1/3}}{p_0^{2/3}}\right)$ thus proving the theorem. □

5 LOWER BOUNDS

In this section, we show that the algorithms for discrete and continuous bid spaces are near-optimal. We start with a lower bound for

discrete bid space as a function of m and OPT. The proofs appear in the appendix.

THEOREM 5.1. *For discrete bid-spaces, there exists an instance I such that any algorithm will incur a regret of at least $\Omega\left(\sqrt{mOPT}\right)$.*

PROOF SKETCH. The lower bound follows by adapting the classical lower bound for stochastic bandits [7]. We consider one arm for each platform and a time horizon $T = 2B$. Each platform $i \in [m]$ different from j has expected reward $r_i = 1/2$ and fixed cost $c_i = 1/2$, while platform j has expected reward $r_j = 1/2(1 + \epsilon)$ with fixed cost $c_j = 1/2$. Each platform needs to be executed $1/\epsilon^2$ times in order to find out the best arm. The total budget needed in order to find the best arm is therefore equal to B. The regret for all arms is $\Omega(m/\epsilon)$. By setting $\epsilon = \sqrt{m/B}$ we obtain the lower bound since the optimal stochastic policy will play arm j for all the $2B$ rounds with cost B and optimal reward $OPT = (1 + \epsilon)B$. □

THEOREM 5.2. *For continuous bid spaces, there exists an instance I such that any algorithm will incur a regret of at least $\Omega\left(m^{1/3}B^{2/3}\right)$.*

PROOF SKETCH. We start with the simple case of $m = 1$. The proof of this theorem is derived by using the lower-bound construction for for Lipshitz bandits (see Chapter 4 in [35]). In particular, we consider the simplest setting of 1-platform with no restrictions on the budget constraint (*i.e.*, $B = T$). The objective function of the algorithm thus, is to maximize the function $f(p)$, where f is the continuous density function of the cumulative distribution of the critical bids P. Since, this is continuous this also implies that this function is 1-Lipshitz and thus, the setting reduces to that of Lipshitz bandits. From Theorem 4.2 in [35] we have that any algorithm incurs a regret of at-least $\Omega(T^{2/3})$.

Consider the case of m platforms. The proof uses a similar strategy as in the lower bound for Lipshitz bandits (see Chapter 4 in [35]) combined with the lower-bound strategy used for semi-bandits (*e.g.*, section 6 in [28]). We will closely follow the notations used in [35]. Define the instance $I(x^*, \epsilon)$ by the following. $\mu(x, i)$ denotes the mean reward for arm $x \in [0, 1]$ for platform $i \in [m]$. In a given instance all the platforms have the same mean reward function.

$$
\mu(x, i) = \begin{cases} \frac{1}{2} & \text{all arms x such that } |x - x^*| \geq \epsilon \\ \frac{1}{2} + \epsilon - |x - x^*| & \text{otherwise} \end{cases}
$$

Similar to [35], we will now construct instances $\mathcal{J}(a^*, \epsilon)$ which is a semi-bandit problem on finite number of arms. More precisely, fix $K \in \mathbb{N}$ to be fixed in the analysis. We define a semi-bandit problem on $K * m$ atoms indexed as (k_i, m_j) for $i \in [K]$ and $j \in [m]$. At each time-step the algorithm can choose exactly one of the atoms from the subset of atoms $\{(k_i, m_j)\}_{i \in [K]}$ for each $j \in [m]$. Thus, at each time-step the algorithm chooses atmost m atoms. For any fixed k_i for $i \in [K]$, the reward assigned for the atoms $\{(k_i, m_j)\}_{j \in [m]}$ is the same and set as in the lower-bound proof for Lipshitz bandits [35].

We now use the observation made in [28]; an instance with m copies of a K-armed bandit problem (*i.e.*, each arm is copied m times) is equivalent to a single K-armed bandit problem where the reward is scaled by a factor m. Thus, the instance $\mathcal{J}(a^*, \epsilon)$ can be replaced by another instance $\mathcal{J}'(a^*, \epsilon)$ on K arms, such that

the reward is scaled by a factor m. This leads us to the following equivalent version of Theorem 4.2 in [35].

THEOREM 5.3 (THEOREM 4.2 FROM [35]). *For a stochastic multi-armed bandit problem with K arms and time-horizon, with rewards in the range $[0, m]$. Let ALG be any algorithm for this problem. Pick a positive constant $\epsilon \leq \sqrt{\frac{cmK}{T}}$ where c is some absolute constant. Then there exists an instant $\mathcal{J}' = \mathcal{J}'(a^*, \epsilon)$ for $a^* \in [K]$ such that*

$$\mathbb{E}[R(T) \mid \mathcal{J}'] \geq \Omega(\epsilon T).$$

Thus, choosing $\epsilon = \frac{1}{2K}$ and $K = \left(\frac{T}{cm}\right)^{1/3}$ and proceeding as in [35] we obtain a regret lower-bound of $\Omega(m^{1/3}T^{2/3})$ as claimed in the theorem. □

6 EXPERIMENTS

6.1 Setup

We construct a real-world dataset obtained from the logs of a large internet advertising company. In this dataset, we have 10 platforms and 3 budget constrained advertisers. For each advertiser, we normalize the dataset and obtain the relevant distributions for the critical bid P and the valuation V. Our final dataset is obtained by sampling the critical bids from P and the valuations from V at each time-step. Due to the nature of the bidding system, we have a good approximation to the *true valuations* for each advertiser on each platform. Outside of the larger advertisers, most advertisers have much smaller budgets compared to their total potential audience and thus, the most interesting regime is when $B \ll T$.

We run the various algorithms for $T = 10^5$ time-steps with a pre-determined discretization of the bid space. Due to the nature of the price distributions for the various platforms (see Figure 1 for representative distributions), we choose the ϵ-hyperbolic mesh as the discretization, where the bid is of the form $\frac{1}{1+\epsilon \cdot \ell}$ where $\ell \in \mathbb{N}$. This is similar to the discretization used for dynamic procurement in [11]. We vary the budget B and compare the total reward obtained by the different algorithms against the optimal *LP* value computed on the mean valuation and prices. For each setting of the parameters, we run 5 independent runs and compute the average as the reward. Additionally, we also measure the run-time of our algorithm and compare that against the baseline LuekerLearn algorithm.

6.2 Algorithms

We compare our algorithm against three different baselines. The first baseline is the SemiBwK-RRS algorithm proposed in [34]. As mentioned in the introduction, on the theoretical front, our algorithm improves over this algorithm in worst-case scenario when $OPT << T$ and when $B << \sqrt{T}$. Thus, we expect to improve over this algorithm in this regime, while having similar performance in the large budget regime. The second baseline is the naive UCB algorithm that ignores the budget constraints and maximizes the rewards. The third algorithm is the *The LuekerLearn Algorithm* from [37] adapted to multiple resources and valuation function in the objective. In particular, we run m different copies of this algorithm, one for each platform. At each time-step, we divide the total remaining budget uniformly across the m different platforms. In other words, if the remaining budget at time t is B_t, each instance of the

LuekerLearn algorithm will receive the remaining budget as B_t/m. Thus, after each time-step, each instance of the algorithm has a *synchronization* where unused budget from one platform can be *transferred* to the other platforms. The full algorithm is described in the Appendix. We would like to emphasize that the baselines perform almost as good as the algorithm proposed in this paper when m is small (see Fig. 3), thus, suggesting that the modified LuekerLearn algorithm is a non-trivial baseline.

6.3 Results

Fig. 2 shows the variation of the total reward collected by each of the three algorithms as a function of budget B for a given advertiser with $T = 10^5$ steps. We see that when the total budget becomes larger and a constant fraction of T, all algorithms perform nearly well. When the budget becomes smaller, both the UCB algorithm and the modified LuekerLearn algorithm collects lower total reward since they run out of budget very early. Moreover, we see that our algorithm performs better than SemiBwK-RRS in this regime and matches the theory. When the budget increases, both our algorithm and SemiBwK-RRS have comparable performance. In Fig. 3, we study the effect of the number of platforms on the total reward. We randomly choose a subset $m \subseteq [10]$ and run the three algorithms (with $B = 10^3$ and $T = 10^5$). We see that with fewer platforms, the difference between the modified LuekerLearn algorithm and our algorithm vanishes. Addtionally, we also see that our algorithm performs slightly better than SemiBwK-RRS when the number of platforms are large. This is unexplained by theory but seems to hold empirically. In Fig. 4, we look at the average stopping time of the various algorithms. We can see that our algorithm (also SemiBwK-RRS which we omit for clarity) depletes the budget uniformly and runs till the end (*i.e.,* $T = 10^5$) while both the baselines deplete their budget within a small fraction of the total time-steps. In advertising platforms, advertisers expect the budget to be used up uniformly over a large period (*e.g.,* a day). Thus, algorithms that deplete the budget very quickly are not desirable, even if they end up collecting larger reward. This is another feature of our algorithm that makes it useful for practice.

7 CONCLUSION

In this paper we presented algorithms for budget management in online advertising across m bidding platforms. We modeled the problem as Stochastic Bandits with Knapsack with an m-dimensional bidding vector. We designed an algorithm for bidding in discrete and continuous actions spaces and proved a mathematical bound on it regret. We also sketched a lower-bound to show that this is optimal. Finally, we used real-world datasets to show extensive empirical evaluation and compared against other competitive heuristics. From a practical stand-point, we believe our work can open directions in modeling for bidding in auctions. First, it is a challenging task to extend to settings where the auction results have correlations across different time-steps. Second, we expect the study of pacing strategies for multi platform advertisement to provide challenging problems for practical and theoretical investigations even beyond the multi-armed bandit setting.
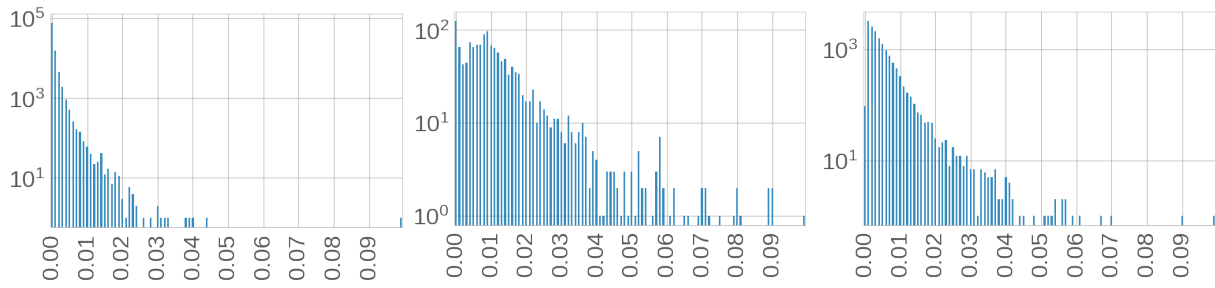
Figure 1: Representative price distribution for three platforms and one advertiser
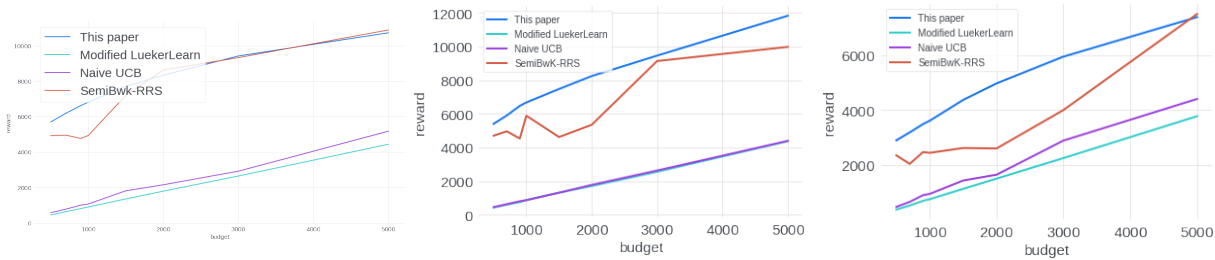

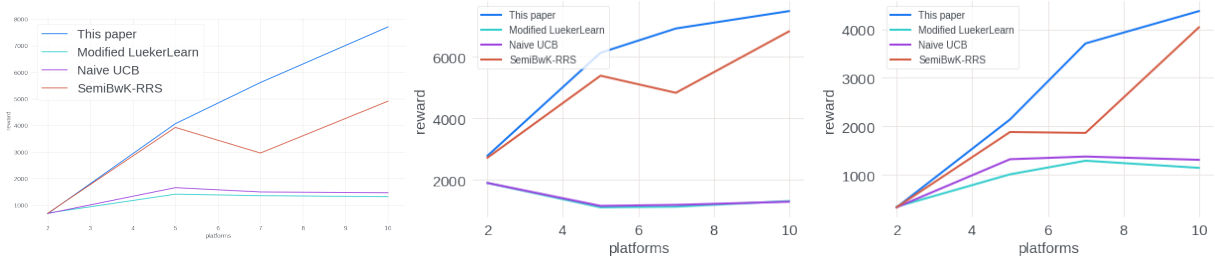Figure 2: Total reward obtained as a function of budget

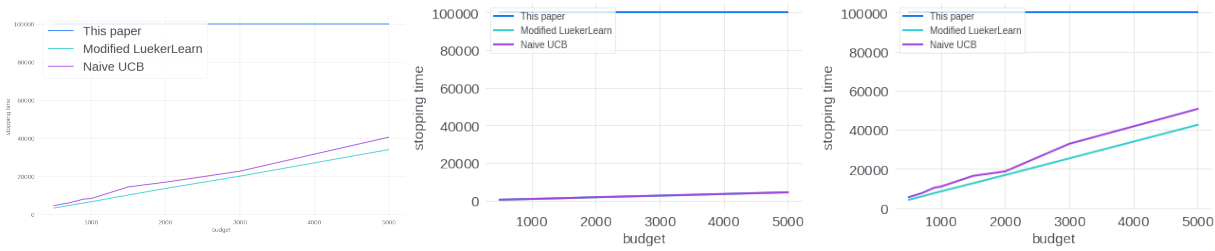Figure 3: Total reward obtained as a function of number of platforms

Figure 4: Average stopping time of the three algorithms as a function of budget

REFERENCES

[1] Zoë Abrams, S Sathiya Keerthi, Ofer Mendelevitch, and John Tomlin. 2008. Ad delivery with budgeted advertisers: A comprehensive LP approach. 9 (01 2008).

[2] Shipra Agrawal and Nikhil R. Devanur. 2014. Bandits with Concave Rewards and Convex Knapsacks. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation (EC '14)*. Association for Computing Machinery, New York, NY, USA, 989–1006. https://doi.org/10.1145/2600057.2602844

[3] Shipra Agrawal and Nikhil R. Devanur. 2016. Linear Contextual Bandits with Knapsacks. In *29th Advances in Neural Information Processing Systems (NIPS)*.

[4] Shipra Agrawal, Nikhil R Devanur, and Lihong Li. 2016. An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. In *Conference on Learning Theory*. 4–18.

[5] Shipra Agrawal, Nikhil R. Devanur, and Lihong Li. 2016. An efficient algorithm for contextual bandits with knapsacks, and an extension to concave objectives. In *29th Conf. on Learning Theory (COLT)*.

[6] Kareem Amin, Michael Kearns, Peter Key, and Anton Schwaighofer. 2012. Budget optimization for sponsored search: Censored learning in MDPs. *arXiv preprint arXiv:1210.4847* (2012).

[7] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. 2002. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.* 32, 1 (2002), 48–77. https://doi.org/10.1137/S0097539701398375 arXiv:https://doi.org/10.1137/S0097539701398375

[8] Vashist Avadhanula, Jalaj Bhandari, Vineet Goyal, and Assaf Zeevi. 2016. On the tightness of an LP relaxation for rational optimization and its applications. *Operations Research Letters* 44, 5 (2016), 612 – 617. https://doi.org/10.1016/j.orl.2016.07.001

[9] Yossi Azar, Benjamin Birnbaum, Anna R. Karlin, and C. Thach Nguyen. 2009. On Revenue Maximization in Second-Price Ad Auctions. In *Algorithms - ESA 2009*, Amos Fiat and Peter Sanders (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 155–166.

[10] A. Badanidiyuru, R. Kleinberg, and A. Slivkins. 2013. Bandits with Knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. 207–216.

[11] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2018. Bandits with Knapsacks. *J. ACM* 65, 3, Article 13 (March 2018), 55 pages. https://doi.org/10.1145/3164539

[12] Ashwinkumar Badanidiyuru, John Langford, and Aleksandrs Slivkins. 2014. Resourceful Contextual Bandits. In *Proceedings of The 27th Conference on Learning Theory (Proceedings of Machine Learning Research)*, Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári (Eds.), Vol. 35. PMLR, Barcelona, Spain, 1109–1134. http://proceedings.mlr.press/v35/badanidiyuru14.html

[13] Santiago Balseiro, Anthony Kim, Mohammad Mahdian, and Vahab Mirrokni. 2017. Budget Management Strategies in Repeated Auctions. In *Proceedings of the 26th International World Wide Web Conference, Perth, Australia*.

[14] 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*.

[15] Matthew Cary, Aparna Das, Benjamin Edelman, Ioannis Giotis, Kurtis Heimerl, Anna R. Karlin, Claire Mathieu, and Michael Schwarz. 2007. Greedy bidding strategies for keyword auctions.. In *EC*, Jeffrey K. MacKie-Mason, David C. Parkes, and Paul Resnick (Eds.). ACM, 262–271. http://dblp.uni-trier.de/db/conf/sigecom/sigecom2007.html#CaryDEGHKMS07

[16] Vincent Conitzer, Christian Kroer, Debmalya Panigrahi, Okke Schrijvers, Eric Sodomka, Nicolas E. Stier-Moses, and Chris Wilkens. 2019. Pacing Equilibrium in First-Price Auction Markets. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC '19)*. ACM, New York, NY, USA, 587–587. https://doi.org/10.1145/3328526.3329600

[17] Vincent Conitzer, Christian Kroer, Eric Sodomka, and Nicolas E. Stier-Moses. 2018. Multiplicative Pacing Equilibria in Auction Markets. In *Conference on Web and Internet Economics (WINE'18)*. Oxford, UK.

[18] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* 97, 1 (2007), 242–259.

[19] Jon Feldman, S Muthukrishnan, Martin Pal, and Cliff Stein. 2007. Budget optimization in search-based advertising auctions. In *Proceedings of the 8th ACM conference on Electronic commerce*.

[20] 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.

[21] Arthur Flajolet and Patrick Jaillet. 2017. Real-time bidding with side information. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc., 5168–5178.

[22] Ashish Goel, Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. 2010. Advertisement Allocation for Generalized Second-pricing Schemes. *Oper. Res. Lett.* 38, 6 (Nov. 2010), 571–576. https://doi.org/10.1016/j.orl.2010.09.002

[23] Kartik Hosanagar and Vadim Cherepanov. 2008. Optimal bidding in stochastic budget constrained slot auctions. In *Proceedings 9th ACM Conference on Electronic Commerce (EC-2008), Chicago, IL, USA, June 8-12, 2008*. 20. https://doi.org/10.1145/1386790.1386794

[24] N. Immorlica, K. A. Sankararaman, R. Schapire, and A. Slivkins. 2019. Adversarial Bandits with Knapsacks. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. 202–219.

[25] Edward L Kaplan and Paul Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American statistical association* 53, 282 (1958), 457–481.

[26] Chinmay Karande, Aranyak Mehta, and Ramakrishnan Srikant. 2013. Optimizing Budget Constrained Spend in Search Advertising. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13)*. ACM, New York, NY, USA, 697–706. https://doi.org/10.1145/2433396.2433483

[27] Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. 2010. Regret bounds for sleeping experts and bandits. *Machine learning* 80, 2-3 (2010), 245–272.

[28] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*. 535–543.

[29] Nick Littlestone and Manfred K. Warmuth. 1994. The Weighted Majority Algorithm. *Inf. Comput.* 108, 2 (Feb. 1994), 212–261. https://doi.org/10.1006/inco.1994.1009

[30] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. 2007. Adwords and generalized online matching. *Journal of the ACM (JACM)* 54, 5 (2007).

[31] Alessandro Nuara, Francesco Trovò, Nicola Gatti, and Marcello Restelli. 2020. Online Joint Bid/Daily Budget Optimization of Internet Advertising Campaigns. arXiv:cs.LG/2003.01452

[32] Raghu Pasupathy and Sujin Kim. 2011. The Stochastic Root-Finding Problem: Overview, Solutions, and Open Questions. *ACM Trans. Model. Comput. Simul.* (2011).

[33] Paat Rusmevichientong and David P. Williamson. 2006. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In *Proceedings 7th ACM Conference on Electronic Commerce (EC-2006), Ann Arbor, Michigan, USA, June 11-15, 2006*. 260–269. https://doi.org/10.1145/1134707.1134736

[34] Karthik Abinav Sankararaman and Aleksandrs Slivkins. 2018. Combinatorial Semi-Bandits with Knapsacks. In *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*. 1760–1770.

[35] Aleksandrs Slivkins. 2019. Introduction to Multi-Armed Bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286. https://doi.org/10.1561/2200000068

[36] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas R Jennings. 2012. Knapsack based optimal policies for budget–limited multi–armed bandits. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[37] Long Tran-Thanh, Lampros Stavrogiannis, Victor Naroditskiy, Valentin Robu, Nicholas R Jennings, and Peter Key. 2014. Efficient Regret Bounds for Online Bid Optimisation in Budget-Limited Sponsored Search Auctions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI'14)*. AUAI Press, Arlington, Virginia, USA, 809–818.

[38] Hal R Varian. 2007. Position auctions. *international Journal of industrial Organization* 25, 6 (2007), 1163–1178.

[39] Donglin Zeng et al. 2004. Estimating marginal survival function by adjusting for dependent censoring using many covariates. *The Annals of Statistics* 32, 4 (2004), 1533–1555.

A MODIFIED LUEKERLEARN ALGORITHM

In this section, we formally describe the modified LuekerLearn algorithm we employ as a baseline in the experimental section. The unbiased maximum likelihood estimate for censored data is given by the Zeng's estimator in Steps 1-5 of [39], which was first used in [37]. A simpler estimate, which is not unbiased, is the famous Kaplan-Meier estimator [25] which was used in [6]. After implementing both, we found that the performance was similar and thus, throughout the experimental section we use the simpler Kaplan-Meier estimator. For our purposes, the Kaplan-Meier estimator is defined as follows. Let $D(t', b, p)$ denote the number of time-steps until t' such that bidding b on platform p did not result in a click (*i.e.*, $b < p$) and let $N(t', b, p)$ denote the number of times until t' we bid b on platform p. Then, the estimate $\hat{p}_t(b, p)$ is defined as

$$1 - \prod_{t'=1}^{t-1} \left(1 - \frac{D(t', b, p)}{N(t', b, p)}\right).$$

Algorithm 2 Modified LuekerLearn algorithm

Initialize $B_1 = B$, $\hat{p}_1(i, b) = 1$ for all platforms $i \in [m]$ and all bids $b \in [0, 1]$.

for $t = 1, \ldots, \tau$ (i.e., until resource budget is exhausted) **do**

For each platform $p \in [m]$ play the bid that maximizes the following

$$j_p^* = \underset{b \in [0,1]}{\arg\max} \qquad\qquad b \qquad\qquad\qquad (6)$$

$$\text{s.t.} \quad \textstyle\sum_{0 \leq \sigma \leq b} \hat{p}_t(p, \sigma)\sigma \leq \frac{B_t}{m(T-t+1)} \qquad (7)$$

Play arm $x_{\boldsymbol{b}_t}$ with bid $\boldsymbol{b}_t = \{b_{p,j_p^*}\}_{p\in[m]}$

Update B_t to be the residual budget

Update $\hat{p}$ for each bid and platform using the unbiased maximum likelihood estimate for censored data.

end for