

# Adaptive Bidding for Display Advertising

Arpita Ghosh  
Yahoo! Research  
2821 Mission College Blvd.  
Santa Clara, CA 95054  
arpita@yahoo-inc.com

Sergei Vassilvitskii  
Yahoo! Research  
2821 Mission College Blvd.  
Santa Clara, CA 95054  
sergei@yahoo-inc.com

Benjamin I. P. Rubinstein  
Computer Science Division  
University of California, Berkeley  
Berkeley, CA 94720  
benr@cs.berkeley.edu

Martin Zinkevich  
Yahoo! Research  
2821 Mission College Blvd.  
Santa Clara, CA 95054  
maz@yahoo-inc.com

## ABSTRACT

Motivated by the emergence of auction-based marketplaces for display ads such as the Right Media Exchange, we study the design of a bidding agent that implements a display advertising campaign by bidding in such a marketplace. The bidding agent must acquire a given number of impressions with a given target spend, when the highest external bid in the marketplace is drawn from an *unknown* distribution  $\mathcal{P}$ . The quantity and spend constraints arise from the fact that display ads are usually sold on a CPM basis. We consider both the full information setting, where the winning price in each auction is announced publicly, and the partially observable setting where only the winner obtains information about the distribution; these differ in the penalty incurred by the agent while attempting to learn the distribution. We provide algorithms for both settings, and prove performance guarantees using bounds on uniform closeness from statistics, and techniques from online learning. We experimentally evaluate these algorithms: both algorithms perform very well with respect to both target quantity and spend; further, our algorithm for the partially observable case performs nearly as well as that for the fully observable setting despite the higher penalty incurred during learning.

## 1. INTRODUCTION

A bidding agent is an entity that implements an online advertising campaign by bidding in ad auctions on behalf of an advertiser. A recent development in Web advertising is the emergence of an *auction-based marketplace* for display ads, where advertisers can bid on individual display advertising opportunities in real time auctions, as in sponsored search. Such a marketplace allows advertisers greater flexibility in the design and implementation of their display advertising campaigns, which were previously restricted to contracts with publishers at pre-negotiated prices. In this paper, we study the design of bidding agents for display advertising, which implement an ad campaign by bidding in such an auction-based marketplace. The bidding agent in question could either be an advertiser herself, or an intermediary acting on behalf of the advertiser.

Copyright is held by the author/owner(s).  
WWW2009, April 20-24, 2009, Madrid, Spain.

Since display advertising is usually sold on a per impression (CPM) basis, a campaign for display advertising has different goals and metrics than one for sponsored search [2]. A CPM-based campaign typically has a target quantity of impressions that need to be acquired over a certain duration with certain targeting characteristics (the targeting can include information both about the webpage on which the ad will appear, as well as the user viewing the page). In addition to the target quantity, a typical constraint in a CPM campaign is a *budget* constraint on the total spend, since payment is made on a CPM rather than a per click (CPC) basis. We will assume that the bidding agent wants to *exhaust*, rather than simply stay within, the allocated budget, for the following reason. Different advertisers often have different pieces of information regarding how valuable a particular user might be, and often a high bid for an impression reflects this information; thus a high price might indicate high value<sup>1</sup>. This also agrees with anecdotal observations that advertisers prefer to exhaust their budgets.

Consider a bidding agent that needs to win  $d$  impressions and has a total budget  $T$ . We assume that the bidding agent knows the total supply  $n$  of impressions satisfying the targeting requirements (this is clearly not true in practice, but a lower bound on the supply can be used instead). Define  $f = d/n$  to be the fraction of the supply that the agent needs to win, and define  $t = T/d$  to be target spend per impression won. We suppose that the highest bids from other bidders are drawn i.i.d.<sup>2</sup> from a distribution with CDF  $\mathcal{P}$ , and that each impression is sold using a second price auction. In general, a given target quantity and spend need not be simultaneously feasible for the distribution  $\mathcal{P}$  – if this happens, we always choose in favor of quantity (our algorithms can be modified to make the opposite choice).

If the distribution  $\mathcal{P}$  is *known* to the bidding agent, then the problem is simple. Let  $z^* = \mathcal{P}^{-1}(f)$  be the bid that would win fraction  $f$  of the supply. Define  $p^*$  such that  $\mathbb{E}_{\mathcal{P}}[X | X \leq p^*] = t$ , that is,  $p^*$  is the bid that achieves the

<sup>1</sup>We assume that an intermediary implementing a campaign on behalf of an advertiser would also like to deliver high-value impressions subject to the chosen budget. Since our algorithms target supply over budget when they are not simultaneously feasible, the intermediary can deliver the cheapest impressions if desired by choosing a small enough budget.

<sup>2</sup>See Section 5 for a discussion.

target spend  $t$  in expectation. If  $z^* \leq p^*$  then bidding  $p^*$  with probability  $A = f/\mathcal{P}(p^*)$  independently on each available impression achieves both the supply and spend targets in expectation. Otherwise, prioritizing supply, we bid  $z^*$  achieving the desired fraction of supply in expectation.

In practice, of course, the distribution  $\mathcal{P}$  is not known to the bidding agent. Our problem is therefore one of *learning* the unknown distribution  $\mathcal{P}$  in order to meet the target quantity and spend constraints. However, learning incurs a penalty, leading to an explore-exploit tradeoff. The nature of the penalty depends on the assumptions made about the extent of information available to the bidding agent. We consider two settings:

- **Fully Observable Exchange.** Here, the winning bid in each auction is announced publicly, so that the bidding agent can learn without placing any bids and therefore expending any budget. The only tradeoff here is between accuracy and length of exploration, which could affect feasibility during exploitation.
- **Partially Observable Exchange.** A harder problem is when the winning bid is not announced publicly—only the winner receives information about the price and therefore the distribution (a realistic setting in online advertising). Thus the bidding agent can infer information about  $\mathcal{P}$  only by bidding high enough to win, i.e., it must pay for every sample it observes from the distribution.

We begin in Section 2 with algorithm *Learn-Then-Bid* for the fully observable case, and provide performance guarantees using non-asymptotic uniform convergence results for empirical processes. While the algorithm is simple—it observes and then bids according to the empirical distribution, the analysis is a useful first step for the partially observable case. A natural improvement is to continue to learn while exploiting; this algorithm indeed outperforms *Learn-Then-Bid* as shown experimentally in Section 4.

In Section 3 we move on to the partially observable case, which is harder since a cost must be paid for every bid that is observed from the distribution. We give an algorithm *Guess-Double-Panic* that explores the distribution gradually based on a guess-then-double pattern, in order to control the spend in the learning phase. A simple guess-then-double algorithm does not suffice since the distribution may have mass concentrated right above a guess, leading to severe overspending upon doubling. We introduce a panic phase to the algorithm to deal with this problem, which limits the overspending and allows for performance guarantees.

In Section 4, we experimentally evaluate these algorithms, as well as some additional heuristics, on realistic data derived from the right media exchange. Both *Learn-Then-Bid* (and its improved version *Learn-While-Bid*) and *Guess-Double-Panic* perform very well for a wide range of target fractions and spends. The experiments also demonstrate that natural heuristics for the partially observable case are indeed inadequate, and are well outperformed by *Guess-Double-Panic*. Most interestingly, the lack of full information is not a severe handicap: *Guess-Double-Panic* does nearly as well as *Learn-Then-Bid* despite access to only partial information.

**Related work:** Although there are many commercial ventures that optimize campaigns on behalf of advertisers,

the design and analysis of bidding agents for online advertising has not received much attention in the research literature. The focus has largely been on bidding agents for sponsored search keyword auctions—for instance, Cary et al [2] propose and analyze a pragmatic bidding agent for sponsored search. Unlike display advertising, the goal there is to choose the optimal (utility maximizing) slot to bid on for each keyword; the authors show that there is a greedy bidding strategy that leads to convergence to equilibrium when all advertisers use the same strategy.

In this paper, we consider approaches inspired by the online learning literature. In particular, our results are similar to those on UCB1 [1] where there is an unknown stationary distribution over events from round to round; we do almost as well asymptotically as we would had the distribution been known in advance. However, although this work has been extended to uncountably infinite action spaces [4], it differs in two major factors: we try to get the correct average behavior (i.e., to behave correctly on average). Second, the expected price obtained on a given round as a function of the bid (roughly analogous to cost in [5]) is not only non-convex, but there is also no pre-determined Lipschitz constant  $K$ , making discretization of the action space unboundedly inefficient as a method of approximation. As an alternative, we focus on the specific properties of the problem to more efficiently explore and exploit.

## 2. FULLY OBSERVABLE EXCHANGE

We first consider a fully observable exchange where the winning bid is revealed after every auction. After describing the algorithm we use the DKW inequality to bound the error on our estimates and the suboptimality of our performance.

*Learn-Then-Bid* takes as input a target fraction  $f$ , spend  $t$ , supply  $n$  and exploration length  $m$ . It explores for  $m$  steps by bidding 0, that is, simply waiting and forms the empirical CDF  $\mathcal{P}_m$  from the observations  $E_i \sim \mathcal{P}$ . It then computes bid value<sup>3</sup>  $P_m^*$  that would achieve the target spend  $t$ , and  $Z_m^*$  that would achieve the necessary fraction, and bids  $P_m^*$  with probability  $A_m$  if it is feasible to achieve both targets on  $\mathcal{P}_m$ , else it bids  $Z_m^*$ .

---

**Algorithm** *Learn-Then-Bid*( $f, t, n, m$ )

- 1: Bid 0 for the first  $m$  opportunities;
  - 2:  $\mathcal{P}_m(x) \leftarrow m^{-1} \sum_{i=1}^m \mathbf{1}[E_i \leq x]$ ;
  - 3:  $P_m^* \leftarrow \inf \{p : \mathbb{E}_{\mathcal{P}_m}[X | X \leq p] \geq t\}$
  - 4:  $A_m \leftarrow \frac{fn}{(n-m)\mathcal{P}_m(P_m^*)}$
  - 5:  $Z_m^* \leftarrow \inf \left\{ z : \mathcal{P}_m(z) \geq \frac{fn}{n-m} \right\}$
  - 6: **if**  $P_m^* \geq Z_m^*$  **then**
  - 7:   **for** opportunities  $i \in \{m+1, \dots, n\}$  **do**
  - 8:     Bid  $P_m^*$  with probability  $A_m$ , and 0 otherwise.
  - 9: **else**
  - 10:   **for** opportunities  $i \in \{m+1, \dots, n\}$  **do** Bid  $Z_m^*$ .
  - 11: **end if**
- 

We prove that the expected future fraction of impressions won and amount spent per impression by *Learn-Then-Bid* converge in probability with rates exponential in the learning phase length.

We assume that the distribution  $\mathcal{P}$  is continuous, strictly monotonic, has support on  $[a, b]$  where  $0 \leq a < b < \infty$  and

<sup>3</sup>We follow the convention that  $\inf \emptyset := +\infty$ .

$\mathcal{P}(a) = 0$ . This implies that  $g(y) = \mathbb{E}_{\mathcal{P}}[X | X \leq y]$  is well-defined, continuous and strictly increasing over  $[a, b]$ , and so  $\mathcal{P}^{-1}$  and  $g^{-1}$  are well-defined also.

In normal scenarios, one needs a small number of impressions and has a small budget. Moreover, the total number of impressions tends to be large. Many of the results we obtain depend upon the problem being feasible after exploration. In Section 4 we show experimentally that  $m$  can be chosen small enough so that the problem remains feasible for most  $f$  and  $t$ .

**DEFINITION 1.** Define  $\gamma = \frac{fn}{n-m}$ . If  $t \leq \mathbb{E}_{\mathcal{P}}[X]$ , define  $p^* = g^{-1}(t)$ . A problem is **feasible after exploration** if  $t \leq \mathbb{E}_{\mathcal{P}}[X]$  and  $\mathcal{P}(p^*) \geq \gamma$ .

Note that this also implies  $\gamma \leq 1$ . We will refer to  $\gamma$  and  $p^*$  throughout this section as defined above.

## 2.1 Measurement Errors

The performance of the algorithm depends primarily upon the accuracy of the exploration phase measurements.

**DEFINITION 2.** For a given  $\epsilon > 0$ , the algorithm has  **$\epsilon$ -accurate observations** if for all  $x \in [a, b]$ :

$$|\mathcal{P}(x) - \mathcal{P}_m(x)| \leq \epsilon. \quad (1)$$

The following is a restatement of the DKW inequality [3]. For the remainder of the analysis, we will condition our results on the observations being  $\epsilon$ -accurate.

**COROLLARY 3.** For  $\epsilon > 0$ , the probability of the algorithm having  $\epsilon$ -accurate observations is greater than or equal to  $1 - 2 \exp(-2m\epsilon^2)$ .

**LEMMA 4.** Given  $\epsilon > 0$ , if the problem is feasible after exploration, and the Learn-Then-Bid algorithm has  $\epsilon$ -accurate observations then

$$|\mathcal{P}(Z_m^*) - \gamma| \leq \epsilon. \quad (2)$$

**PROOF.** We first consider bidding  $Z_m^*$  on each round of the bidding phase; our goal is to prove that the expected fraction won  $\mathcal{P}(Z_m^*)$  is close to the target fraction  $\gamma$  w.h.p. If  $\gamma \leq \epsilon$ , then  $\mathcal{P}(Z_m^*) \geq 0 \geq \gamma - \epsilon$ . If  $\gamma > \epsilon$ , then  $(0, \gamma - \epsilon] \neq \emptyset$ , and for any  $\epsilon' \in (0, \gamma - \epsilon]$ , define  $z^{*-} = \mathcal{P}^{-1}(\gamma - \epsilon - \epsilon')$ . Because the algorithm has  $\epsilon$ -accurate observations:

$$\begin{aligned} \mathcal{P}_m(z^{*-}) &\leq \mathcal{P}(z^{*-}) + \epsilon = \gamma - \epsilon' \\ &< \gamma \leq \mathcal{P}_m(Z_m^*), \end{aligned}$$

and so  $z^{*-} < Z_m^*$  by the monotonicity<sup>4</sup> of  $\mathcal{P}_m$ . Therefore, for all  $\epsilon' \in (0, \gamma - \epsilon]$ ,  $\mathcal{P}(Z_m^*) > \mathcal{P}(z^{*-}) = \gamma - \epsilon - \epsilon'$ , implying  $\mathcal{P}(Z_m^*) \geq \gamma - \epsilon$ . By a similar argument,  $\mathcal{P}(Z_m^*) \leq \gamma + \epsilon$ .  $\square$

## 2.2 Approximation of Target Supply Won

Our first main result states that *Learn-Then-Bid* wins close to  $fn$  opportunities, if the algorithm has  $\epsilon$ -accurate observations and the problem is feasible after exploration. The following lemma is a consequence of a positive partial derivative.

**LEMMA 5.** For  $\gamma, \epsilon > 0$ ,  $x\gamma(x+\epsilon)^{-1}$  is strictly increasing, and  $x\gamma(x-\epsilon)^{-1}$  is strictly decreasing.

<sup>4</sup>It is important that  $\mathcal{P}_m(z^{*-})$  is strictly less than  $\mathcal{P}_m(Z_m^*)$  because  $\mathcal{P}_m$  is weakly monotonic, thus the need for  $\epsilon' > 0$ .

**THEOREM 6.** Given  $\epsilon > 0$ , and  $j > m$ , let  $B_j$  be the  $j$ th bid of the Learn-Then-Bid algorithm. If the problem is feasible after exploration and the algorithm has  $\epsilon$ -accurate observations, then:

$$\gamma - \epsilon \leq \mathbf{E}[\mathcal{P}(B_j) | E_1 \dots E_m] \leq \frac{\gamma - \epsilon}{\gamma - 2\epsilon} \gamma. \quad (3)$$

**PROOF.** From Lemma 4, we know that  $|\mathcal{P}(Z_m^*) - \gamma| \leq \epsilon$ . If  $Z_m^* \geq P_m^*$  and therefore  $B_j = Z_m^*$ , then  $\gamma - \epsilon \leq \mathcal{P}(B_j) \leq \gamma + \epsilon \leq \frac{\gamma - \epsilon}{\gamma - 2\epsilon} \gamma$ . Therefore for the remainder of the proof we assume that  $\mathcal{P}(P_m^*) \geq \mathcal{P}(Z_m^*) \geq \gamma - \epsilon$ .

In this case,  $B_j = P_m^*$  with probability  $A_m$ , and zero otherwise ( $\mathcal{P}(0) = 0$ ). The conditional probability of winning bid  $j$  given the outcome of the exploration phase is  $\mathcal{P}(P_m^*)A_m$ : it is this value we wish to show is close to the target  $\gamma$ :

$$\mathcal{P}(P_m^*)A_m = \frac{\mathcal{P}(P_m^*)}{\mathcal{P}_m(P_m^*)} \gamma \geq \frac{\mathcal{P}(P_m^*)}{\mathcal{P}(P_m^*) + \epsilon} \gamma \geq \frac{\gamma - \epsilon}{\gamma} \gamma.$$

The first equality follows by definition of  $A_m$ , the first inequality follows due to the  $\epsilon$ -accurate observations, and the final inequality is a consequence of Lemma 5 and the inequality  $\mathcal{P}(P_m^*) \geq \mathcal{P}(Z_m^*) \geq \gamma - \epsilon$ . We upper-bound the expected fraction won in a similar fashion:

$$\frac{\mathcal{P}(P_m^*)}{\mathcal{P}_m(P_m^*)} \gamma \leq \frac{\mathcal{P}(P_m^*)}{\mathcal{P}(P_m^*) - \epsilon} \gamma \leq \frac{\gamma - \epsilon}{\gamma - 2\epsilon} \gamma. \quad \square$$

## 2.3 Approximation of Target Spend

First we convert DKW-type uniform closeness of true and empirical CDFs to uniform closeness of expected spend. This “uniformity” is gained by focusing only on the area of interest, bids that obtain at least  $\gamma - \epsilon$  impressions.

For a given  $\epsilon \in (0, \gamma/2)$ , define  $\epsilon^- = \frac{\epsilon b}{(\gamma - \epsilon)(\gamma - 2\epsilon)} + \frac{\epsilon(b-a)}{\gamma - 2\epsilon}$ , and  $\epsilon^+ = \frac{\epsilon b}{\gamma(\gamma - \epsilon)} + \frac{\epsilon(b-a)}{\gamma}$ .

**LEMMA 7.** Given  $\epsilon \in (0, \gamma/2)$ , let  $z^{*-} = \mathcal{P}^{-1}(\gamma - \epsilon)$  and define the function  $g_m(y) = \mathbb{E}_{\mathcal{P}_m}[X | X \leq y]$ . If the algorithm has  $\epsilon$ -accurate observations, then if  $Y \geq z^{*-}$ :

$$g_m(Y) - \epsilon^- \leq g(Y) \leq g_m(Y) + \epsilon^+.$$

**PROOF.** Because the algorithm has  $\epsilon$ -accurate observations, both of the following relations hold<sup>5</sup> for all  $y \in [a, b]$ :

$$\begin{aligned} |\mathcal{P}(y) - \mathcal{P}_m(y)| &\leq \epsilon \\ \left| \int_a^y (1 - \mathcal{P}(x)) dx - \int_a^y (1 - \mathcal{P}_m(x)) dx \right| &\leq \epsilon(b-a). \end{aligned}$$

Lower-bounding  $g_m(Y)$  with the above relations, identity  $\frac{d}{c+\epsilon} = \frac{d}{c} - \frac{d\epsilon}{c(c+\epsilon)}$ , and  $\mathcal{P}(Y) \geq \mathcal{P}(z^{*-}) = \gamma - \epsilon$ :

$$\begin{aligned} g_m(Y) &= \frac{a + \int_a^Y (1 - \mathcal{P}_m(x)) dx}{\mathcal{P}_m(Y)} \\ &\geq \frac{a + \int_a^Y (1 - \mathcal{P}(x)) dx}{\mathcal{P}(Y) + \epsilon} - \frac{\epsilon(b-a)}{\mathcal{P}(Y) + \epsilon} \\ &= g(Y) - \frac{\epsilon(a + \int_a^Y (1 - \mathcal{P}(x)) dx)}{\mathcal{P}(Y)(\mathcal{P}(Y) + \epsilon)} - \frac{\epsilon(b-a)}{\mathcal{P}(Y) + \epsilon} \\ &\geq g(Y) - \frac{\epsilon b}{\gamma(\gamma - \epsilon)} - \frac{\epsilon(b-a)}{\gamma}. \end{aligned} \quad (4)$$

<sup>5</sup>The integrals are well-defined because  $\mathcal{P}$  and  $\mathcal{P}_m$  are bounded, monotonic functions.

Upper-bounding  $g_m(Y)$  proceeds by the same arguments and identity  $\frac{d}{c-\epsilon} = \frac{d}{c} + \frac{d\epsilon}{c(c-\epsilon)}$ .  $\square$

**THEOREM 8.** *Given the problem is feasible and  $\epsilon \in (0, \gamma/2)$ , If the algorithm has  $\epsilon$ -accurate observations,  $j > m$  and  $B_j$  is the  $j$ th bid and  $B_j > 0$ , then:  $t - \epsilon^- \leq g(B_j) \leq t + \epsilon^+$ .*

PROOF.

$$t - \epsilon^- \leq g(\max(Z_m^*, P_m^*)) \leq t + \epsilon^+ . \quad (5)$$

Note that if  $j > m$  and  $B_j > 0$  then  $B_j = g(\max(Z_m^*, P_m^*))$ . As with Lemma 4, we need to analyze values that bracket the value of interest. In particular, the first value of interest is  $z^{*-} = \mathcal{P}^{-1}(\gamma - \epsilon)$ . If  $g(z^{*-}) \geq t - \epsilon^-$ , then by Lemma 4,  $\max(Z_m^*, P_m^*) \geq Z_m^* \geq z^{*-}$ , and by the monotonicity of  $g$ ,  $g(\max(Z_m^*, P_m^*)) \geq g(z^{*-}) \geq t - \epsilon^-$ . Therefore, we can assume that  $g(z^{*-}) < t - \epsilon^-$ . Since  $t > t - \epsilon^- > g(z^{*-}) \geq a$ ,  $t - \epsilon^-$  is in the range of  $g$  and  $g^{-1}(t - \epsilon^-) \geq z^{*-}$ . For  $\epsilon' \in (0, t - \epsilon^- - a]$ , define  $p^{*-} = g^{-1}(t - \epsilon^- - \epsilon')$ .

$$\begin{aligned} g_m(p^{*-}) &\leq g(p^{*-}) - \epsilon = t - \epsilon' \\ &< t \leq g_m(P_m^*) , \end{aligned}$$

so  $p^{*-} \leq P_m^*$ . Therefore, because  $g$  is monotonically increasing, for any  $\epsilon' > 0$ ,  $t - \epsilon^- - \epsilon' \leq g(P_m^*)$ , implying  $t - \epsilon^- \leq g(P_m^*) \leq g(\max(Z_m^*, P_m^*))$ . By a similar argument,  $t + \epsilon^+ \geq g(P_m^*)$ .  $\square$

From Corollary 3 the algorithm has  $\epsilon$ -accurate observations with probability  $1 - 2 \exp(-2n\epsilon^2)$ . From Theorem 6 and Theorem 8:

**THEOREM 9.** *Given a problem that is feasible after exploration. If  $\epsilon \in (0, \gamma/2)$ : With probability  $1 - 2 \exp(-2n\epsilon^2)$ :*

$$t - \epsilon^- \leq g(B_j) \leq t + \epsilon^+ \text{ if } B_j > 0 \quad (6)$$

$$\gamma - \epsilon \leq \mathbf{E}[\mathcal{P}(B_j) | e_1 \dots e_m] \leq \frac{\gamma - \epsilon}{\gamma - 2\epsilon} \gamma . \quad (7)$$

### 3. PARTIALLY OBSERVABLE EXCHANGE

We now move on to the partially observable case, where information is revealed only to the winner of an auction. This problem is more difficult because the bidder must pay a cost in order to obtain information; specifically, we cannot simply learn about the auction by bidding zero for a while. The most brute force approach is to bid infinity for an exploration period, but that can cause overspending by almost  $b/t$  when the target fraction is small. In this section, we will try to be approximately optimal. If  $b/t$  is small then we can just bid  $b$ : however, we want to handle the case where  $b/t$  is large as well.

Consider a simple algorithm that works rather well: bid  $2t$  blindly until the correct number of impressions are obtained. Observe that since the minimum bid to get  $fn$  impressions is less than the bid that gets an expected price of  $t$ , then the expected price paid when a bid is made which obtains  $fn$  is less than or equal to  $t$ . Therefore, by Markov's inequality, the number of the lowest  $fn$  impressions below  $2t$  is  $fn/2$ . Moreover, the most that one can spend is  $2tfn$ , and since the budget is  $tfn$ , this is only twice the budget. Therefore, this algorithm will not dramatically overspend and it will obtain half the required impressions.

Instead of either of these extremes (bidding  $2t$  blindly or aggressively exploring with  $b$ ), we will apply the guess-then-double pattern. The guess then double approach is used

in a variety of domains. For instance, if we want to create an array of items but do not know how many elements it will contain, we make a guess, and if we need more space, we double the size of the array. Thus, the number of new allocations is logarithmic in the number of elements entered, and the number of copies is linear, and the size may be off by no more than a factor of two. Of course, doubling is only one possibility: in the case of the array, multiplying by a factor smaller than 2 will result in more copies but greater efficiency in space.

In the *Guess-Double-Panic* algorithm below, we apply a modification of this technique to our bids during the exploration phase. We will refer to a bid being used during exploration as an exploration bid. A ‘‘safe’’ bid<sup>6</sup> is  $t$ , the target spend, since there is no danger of going over budget. From this point, we exponentially increase our exploration bid, exploring enough with each new bid to learn the distribution below this bid. At some point we notice our exploration bid exceeds  $p^*$ .

A naive approach is to simply test for this condition at each iteration, and then react to our experience at the end of each phase of the exploration (i.e., remove lines 8-9). However, this can result in a large amount of overspending, as the following example shows<sup>7</sup>. The target price is 10 cents and we need to obtain 10% of the 1000 impressions. Therefore, we are searching for 100 impressions for \$10.00 total. The distribution of bids is: 9.9% are at 1 cent. 0.1% are at \$9.01. Finally, the remaining 90% of the bids are at \$10.00. An ideal bid is \$9.01. However, it is difficult to say whether this one bid will be observed. Unless there is a very slow exploration, the bids will likely exceed \$10.00 during exploration. Thus, there will be a high penalty where the algorithm will most likely pay 10 times its budget.

Instead, if we start overspending during the exploration phase, we go to the Panic() subroutine, where we move into a new phase of ‘‘cautious exploration’’. Effectively, if we are overspending, then we momentarily ignore the budget, and target solely the number of impressions obtained. As before, we continue to increase the bids, but if we realize that we can get enough impressions in the exploitation phase, we immediately move on to the exploitation phase.

The algorithm continues to explore (or panic) until it finds a price  $B_{i^*}$  where it estimates its budget can be exhausted and it can win enough impressions (or it has explored at  $b$  or above). As with the observable case, we have a good approximation of the outcome of bidding any price below  $B_{i^*}$  as we leave the exploration phase.

In particular, there are now three modes of operation. The first mode is *exploration*: the algorithm explores until it either gets enough information or the budget becomes tight. The second (optional) mode is *panic*: the budget is tight, but the algorithm does not have enough information to obtain the right number of impressions, so it aggressively grabs impressions until it reaches a more stable scenario. The third stage is *exploitation*. In the exploitation stage, if there is sufficient budget to obtain the right number of impressions, then the algorithm tries to exhaust the budget. Otherwise, it is cheap and tries to get the right number of

<sup>6</sup>There is also the possibility that we *underspend*. This can happen if we get many opportunities while we are bidding too low. However, this is not a problem so long as the exploration period is sufficiently short with respect to  $\gamma n$ .

<sup>7</sup>A similar continuous distribution is easy to construct.

impressions at a discount price.

For the following algorithm, all variables are global.

---

**Algorithm** *Guess-Double-Panic*( $f, t, n, m, \phi$ )

```

1: Initialize:  $g_{remain} \leftarrow fn$ ;  $budget \leftarrow tg_{remain}$ 
2: Initialize:  $P_0 \leftarrow 0$ ;  $T_0 \leftarrow 0$ ;  $B_0 \leftarrow 0$ ;  $i \leftarrow 0$ ;  $j \leftarrow 0$ 
3: while ( $T_i g_{remain} < budget$  or  $P_i(n-j) < g_{remain}$ ) and  $B_i < b$  do
4:    $i \leftarrow i + 1$ .
5:    $B_i \leftarrow t(\phi)^{i-1}$ .
6:    $S_i \leftarrow \emptyset$  to be a multiset.
7:   for  $k = 1$  to  $m$  do
8:     if  $(g_{remain} - 1)T_{i-1} + B_{i-1} > budget$  then
9:       return Panic()
10:     $j \leftarrow j + 1$ .
11:    if  $g_{remain} = 0$  then Terminate.
12:    Bid  $B_i$ .
13:    if Bid wins then
14:      Define  $p_j \leftarrow$  price won.
15:       $g_{remain} \leftarrow g_{remain} - 1$ .
16:       $budget \leftarrow budget - p_j$ .
17:      Add  $p_j$  to  $S_i$ .
18:    end if
19:  end for
20:  if  $S_i \neq \emptyset$  then  $T_i \leftarrow \frac{1}{|S_i|} \sum_{p \in S_i} p$  else  $T_i \leftarrow 0$ .
21:   $P_i \leftarrow \frac{|S_i|}{m}$ .
22: end while
23:  $i^* \leftarrow i$ .
24: return Exploit().

```

---

The first danger of using any exploration technique is that the problem may be unsolvable if one spends too much time exploring. Note that there will be no more than  $r = \lceil \log_\phi(b/t) \rceil + 1$  rounds of exploration, because then the bid will be above  $b$ . In each round, there are  $m$  bids, so  $mr$  is the maximum number of steps of exploration.

DEFINITION 10. Define  $\gamma = \frac{fn}{n-mr}$ . If  $t \leq \mathbf{E}_{\mathcal{P}}[X]$ , define  $p^* = g^{-1}(t)$ . A problem is **feasible after exploration** if  $t \leq \mathbf{E}_{\mathcal{P}}[X]$  and  $\mathcal{P}(p^*) \geq \gamma$ .

Again, this implies  $\gamma \leq 1$ .

Let us consider the period that generated  $S_i$ . Define  $S_i^*$  to be the multiset of *all* prices (observed and unobserved) during this period (clearly not an observed set). As in the observed case, there are several observations we could make about this distribution (although in this case only theoretically). Formally, define:

$$\mathcal{P}_m^i(x) = m^{-1} |\{y \in S_i^* | y \leq x\}|. \quad (8)$$

Moreover, we can look at these independently from the algorithm itself.

DEFINITION 11. The algorithm has  **$\epsilon$ -accurate observations** if for each  $i \in \{1 \dots r\}$ ,  $\arg\max_{x \in [a, b]} |\mathcal{P}_m^i(x) - \mathcal{P}(x)| \leq \epsilon$ .

LEMMA 12. The probability of not having  $\epsilon$ -accurate observations is less than or equal to  $2(\lceil \log_\phi(b/t) \rceil + 1) \exp(-2m\epsilon^2)$ .

The probability of  $\epsilon$ -accurate observations can be determined by applying the DKW inequality to each round of exploration/panicking, and then applying a union bound.

---

**Subroutine** Panic

```

25: while  $B_{i-1} < b$  do
26:   while  $k \leq m$  do
27:     if  $g_{remain} < P_{i-1}(n-j)$  then
28:        $i^* \leftarrow i - 1$ .
29:       return Exploit().
30:     end if
31:      $j \leftarrow j + 1$ .
32:     Bid  $B_i$ .
33:     if Bid wins then
34:       Define  $p_j \leftarrow$  price won.
35:        $g_{remain} \leftarrow g_{remain} - 1$ .
36:        $budget \leftarrow budget - p_j$ .
37:       Add  $p_j$  to  $S_i$ .
38:     end if
39:      $k \leftarrow k + 1$ .
40:   end while
41:    $k \leftarrow 1$ .
42:   if  $S_i \neq \emptyset$  then  $T_i \leftarrow \frac{1}{|S_i|} \sum_{p \in S_i} p$  else  $T_i \leftarrow 0$ .
43:    $P_i \leftarrow \frac{|S_i|}{m}$ .
44:    $i \leftarrow i + 1$ .
45:    $B_i = t(\phi)^{i-1}$ .
46:    $S_i \leftarrow \emptyset$  to be a multiset.
47: end while
48:  $i^* \leftarrow i - 1$ .
49: return Exploit().

```

---



---

**Subroutine** Exploit

```

50: If  $g_{remain} \leq 0$  or  $j \geq n$  then Terminate.
51:  $B_{\text{final}} \leftarrow B_{i^*}$ .
52:  $A \leftarrow \frac{g_{remain}}{P_{i^*}(n-j)}$ .
53: if  $P_{i^*}(n-j) > g_{remain}$  and  $T_{i^*} g_{remain} > budget$  then
54:   Sort  $p \in S_{i^*}$ : define  $q_k$  to be the  $k$ th smallest  $p$  in  $S_{i^*}$ .
55:    $k_s \leftarrow \left\lceil \frac{g_{remain} m}{n-j} \right\rceil$ 
56:   for  $k = 1$  to  $|S_{i^*}|$  do  $g_k \leftarrow \frac{1}{k} \sum_{i=1}^k q_i$ .
57:    $t^* \leftarrow \frac{budget}{g_{remain}}$ .
58:    $k_p \leftarrow \min_{k: g_k \geq t^*} k$ .
59:    $k^* \leftarrow \max(k_s, k_p)$ .
60:    $B_{\text{final}} \leftarrow q_{k^*}$ .
61:    $A \leftarrow \frac{k^*}{m}$ .
62:    $A \leftarrow \frac{g_{remain}}{A(n-j)}$ .
63: end if
64: while More rounds and  $g_{remain} > 0$  do
65:   Bid  $B_{\text{final}}$  with probability  $A$ , 0 otherwise.
66:   If Bid won then  $g_{remain} \leftarrow g_{remain} - 1$ .
67: end while

```

---

Define  $C_{\text{explore}}$ ,  $C_{\text{panic}}$ , and  $C_{\text{exploit}}$  to be the spend during exploration, panic, and exploitation, respectively. Define  $n_{\text{explore}}$ ,  $n_{\text{panic}}$ , and  $n_{\text{exploit}}$  to be the number of impressions won during exploration, panic, and exploitation phases, respectively. Define  $m'$  to be the number of opportunities during the exploration and panic phases combined. Define  $n_{\text{exploit}}^*$  to be the target number of impressions that remain after the exploration and panic phases. Note that this is equal to  $g_{\text{remain}}$  at the beginning of the exploitation phase. Instead of targeting  $\gamma$  as we did before, we are targeting  $\gamma' = \frac{n_{\text{exploit}}^*}{n - m'}$ , where  $\gamma' \leq \gamma$  because  $n_{\text{exploit}}^* \leq fn$  and  $m' \leq mr$ , implying that  $\gamma' \leq 1$ . Because the number of impressions obtained is a priority, the proof that this is achieved is fairly straightforward. The spend we will bound is  $C = C_{\text{explore}} + C_{\text{panic}} + \mathbf{E}[C_{\text{exploit}} | E_1 \dots E_{m'}]$ .

**THEOREM 13.** *If the problem is feasible after exploration, and the algorithm has  $\epsilon$ -accurate observations, then:*

$$fn \geq n_{\text{explore}} + n_{\text{panic}} + (n - m')\mathcal{P}(B_{\text{final}})A \geq fn - \epsilon,$$

where  $B_{\text{final}}$  and  $A$  are as in Line 65.

**PROOF.** The upper bound is true because there is always a check that  $g_{\text{remain}} > 0$  before any bid is made. The proof of the lower bound has the same rough outline as Theorem 6.  $q_{k_s}$  is analogous to  $Z_m^*$ , and  $q_{k_p}$  is analogous to  $P_m^*$ . The fundamental difference between this and the proof of Theorem 6 is that during the exploration phase, the target fraction of impressions to win during exploitation may be lower than  $\gamma$ , due to some being won during the exploration and panic. However, it is still easy to prove that  $|\mathcal{P}(q_{k_s}) - \gamma'| \leq \epsilon$  using techniques similar to Lemma 4, because the proof does not depend upon the target. The most serious issue is when  $k_p \geq k_s$ , (analogous to  $P_m^* \geq Z_m^*$ ), there is no implicit lower bound on  $\mathcal{P}(q_{k_p})$  outside of  $\gamma'$ . This makes the upper bounds that we obtained in Theorem 6 impossible to obtain, and why we explicitly bound the number of impressions obtained from above. However, the lower bounds on  $A\mathcal{P}(q_{k_p})$  work out just as in Theorem 6.  $\square$

### 3.1 Bounding the Amount Spent

We prove an upper bound on the spend of the algorithm.

**THEOREM 14.** *If the algorithm has  $\epsilon$ -accurate observations and the problem is feasible after exploration, then  $C \leq \phi tfn + (\epsilon^{1/3} + \epsilon^{2/3})nb$ .*

**PROOF.** With Lemmas 15-21, we cover overspending based upon every outcome of exploration and panic, as well as the relationship between  $k_s$  and  $k_p$ . In particular:

1. If the algorithm exits exploration on Line 24, then Lemma 15 applies if  $k_p \geq k_s$ , and Lemma 21 applies if  $k_p < k_s$ .
2. If the algorithm exits exploration on Line 11, then Lemma 16 applies.
3. If the algorithm exits exploration on Line 9, then Lemma 17 applies if no panic bids are made, Lemma 18 applies if a panic bid of  $B_{i^*+1}$  is made, Lemma 20 applies if the largest panic bid is  $B_{i^*}$  and  $k_p \leq k_s$ , and Lemma 19 applies if  $k_p > k_s$ .  $\square$

**LEMMA 15.** *If the algorithm exits the exploration phase at Line 24, if the algorithm has  $\epsilon$ -accurate observations, the problem is feasible after exploration, and  $k_p \geq k_s$ , then the amount overspent is less than  $C \leq tfn + (\epsilon^{1/3} + \epsilon^{2/3})bn$ .*

**PROOF.** As before, if a large fraction of the impressions remains, then the estimates will be pretty accurate. On the other hand, if many of the impressions are already gone, the impact of making a mistake is less.

Basically, we choose a point,  $\gamma^* = \epsilon^{1/3} + \epsilon$ , as a threshold: if  $\gamma' > \gamma^*$ , we can bound the spend normally. If  $\gamma' \leq \gamma^*$ , then the most we can spend is  $\gamma^*b$ .

As we consider the bound proven in Lemma 7, if we replace  $\gamma$  with  $\gamma^*$  in Equation 4, then we get:

$$g_m(Y) \geq g(Y) - \frac{\epsilon b}{\gamma^*(\gamma^* - \epsilon)} - \frac{\epsilon(b-a)}{\gamma^*} \quad (9)$$

$$= g(Y) - \frac{\epsilon b}{(\epsilon + \epsilon^{1/3})(\epsilon^{1/3})} - \frac{\epsilon(b-a)}{\epsilon^{1/3}} \quad (10)$$

$$\geq g(Y) - \epsilon^{1/3}b - \epsilon^{2/3}(b-a) \quad (11)$$

$$\geq g(Y) - \epsilon^{1/3}b - \epsilon^{2/3}b. \quad (12)$$

Equation 12 follows because decreasing the denominator (replacing  $\epsilon + \epsilon^{1/3}$  with  $\epsilon$ ) makes the term larger, but making a negative term larger makes the overall expression smaller. Also, since  $\epsilon \leq 1$ ,  $\epsilon^{2/3} \geq \epsilon$ , implying  $\gamma^*b \leq \epsilon^{1/3}b + \epsilon^{2/3}b$ .  $\square$

**LEMMA 16.** *If exploration exits due to Line 11, then  $C \leq \phi tfn$ .*

**PROOF.** Define  $B^*$  to be equal to *budget* at the last time Line 8 was visited. This is the ideal amount to bid on the last bid such that the budget is exactly met. The maximum amount that could be bid would be  $B_i$ . Before the last bid was made,  $g_{\text{remain}} = 1$  and (due to Line 8),  $B_{i-1} \leq B^*$ . Therefore,  $B_i \leq \phi B^*$ . The maximum amount overspent would be  $B_i - B^* \leq (\phi - 1)B^*$ . Since  $B^* \leq tfn$ , the result follows.  $\square$

**LEMMA 17.** *If exploration exits on Line 9, and no panic bids (Line 32) are made, then  $C \leq \phi tfn + (\epsilon^{1/3} + \epsilon^{2/3})nb$ .*

**PROOF.** During the middle of an exploration phase, an impression is won (which we will call the **overpriced impression**) and Line 8 is true. Define  $p_{\text{op}}$  to be the price of this bid. At the time when the expression in Line 8 is true:

$$tfn - \text{budget} = C_{\text{explore}} + C_{\text{panic}} \quad (13)$$

$$g_{\text{remain}} = n_{\text{exploit}}^* \quad (14)$$

Consider the last time that Line 8 is false, i.e. when one less impression was won and  $p_{\text{op}}$  less was spent.

$$tfn - \text{budget} = C_{\text{explore}} + C_{\text{panic}} - p_{\text{op}} \quad (15)$$

$$g_{\text{remain}} - 1 = n_{\text{exploit}}^* \quad (16)$$

$$(g_{\text{remain}} - 1)T_{i-1} + B_{i-1} \leq \text{budget} \quad (17)$$

$$(n_{\text{exploit}}^*)T_{i-1} + B_{i-1} \leq tfn + p_{\text{op}} - (C_{\text{explore}} + C_{\text{panic}}) \quad (18)$$

In this case, since we go through panic, at this particular time  $i - 1 = i^*$ , we can reframe this as a bound on the "spend":

$$C_{\text{explore}} + C_{\text{panic}} + (n_{\text{exploit}}^*)T_{i^*} \leq tfn + p_{\text{op}} - B_{i^*}.$$

Because  $p_{\text{op}} - B_{i^*} \leq (\phi - 1)B_{i^*} \leq (\phi - 1)tfn$ :

$$C_{\text{explore}} + C_{\text{panic}} + (n_{\text{exploit}}^*)T_{i^*} \leq \phi tfn. \quad (19)$$

The final issue is that  $T_{i^*}$  is not necessarily equal to  $g(B_{i^*})$  (even though  $T_{i^*}$  is an estimate of  $g(B_{i^*})$ ). The remainder is

similar to Lemma 15. There are two scenarios: either  $P_{i^*}$  is small, and therefore few impressions are left to obtain, or  $P_{i^*}$  is large and  $T_{i^*}$  is accurate. Formally, consider  $\gamma^* = \epsilon + \epsilon^{1/3}$ . If  $P_{i^*} \leq \gamma^*$ , then because the expression in Line 27 is true,  $n_{\text{exploit}}^* \leq n\gamma^*$ . If  $P_{i^*} \geq \gamma^*$ , then  $\mathcal{P}(B_{i^*}) \geq \gamma^* - \epsilon$ , then substituting  $\gamma^*$  for  $\gamma$  in Lemma 7:

$$g(B_{i^*}) \leq T_{i^*} + \frac{\epsilon b}{\gamma^*(\gamma^* - \epsilon)} + \frac{\epsilon(b-a)}{\gamma^*} \quad (20)$$

$$= T_{i^*} + \frac{\epsilon b}{(\epsilon^{1/3} + \epsilon)(\epsilon^{1/3})} + \frac{\epsilon(b-a)}{\epsilon + \epsilon^{1/3}} \quad (21)$$

$$\leq T_{i^*} + \epsilon^{1/3}b + \epsilon^{2/3}(b-a) \quad (22)$$

$$\leq T_{i^*} + \epsilon^{1/3}b + \epsilon^{2/3}b. \quad (23)$$

Summarizing:

$$g(B_{i^*})(n_{\text{exploit}}^*) \leq \max(\gamma^*nb, (T_{i^*} + \epsilon^{1/3}b + \epsilon^{2/3}b)fn)$$

$$C_{\text{explore}} + C_{\text{panic}} + n_{\text{exploit}}^*g(B_{i^*}) \leq \phi tfn + \max(\gamma^*nb, (\epsilon^{1/3}b + \epsilon^{2/3}b)fn)$$

Finally, because  $\gamma^* = \epsilon + \epsilon^{1/3}$ , and  $f \leq 1$ , the result follows.  $\square$

LEMMA 18. *If, during the Panic() algorithm, a bid of  $B_{i^*+1}$  is the highest bid that is actually made, the problem is feasible after exploration, and the algorithm has  $\epsilon$ -accurate observations, then  $C \leq fn(t\phi + \epsilon b)$ .*

PROOF. During exploitation, the algorithm bids no more than  $B_{i^*}$ , so the expected price per impression is no more than  $g(B_{i^*})$ . This is the easy part: the hard part is bounding the spend during exploration and panic. Define  $b^* = \min(B_{i^*+1}, b)$  which is the largest effective bid during panicking. How many impressions need to be obtained before the algorithm realizes it can bid  $B_{i^*}$  for the remainder? Formally, this is:

$$\begin{aligned} \max \# \text{big bids won} &= fn - (\mathcal{P}(B_{i^*}) - \epsilon)(n - mr) \\ &= fn \left( 1 - \frac{(\mathcal{P}(B_{i^*}) - \epsilon)(n - mr)}{fn} \right) \\ &= fn \left( 1 - \frac{(\mathcal{P}(B_{i^*}) - \epsilon)}{\gamma} \right) \\ &= fn \left( \frac{\gamma - (\mathcal{P}(B_{i^*}) - \epsilon)}{\gamma} \right) \\ C_{\text{panic}} + C_{\text{exploit}} &\leq fn \left( \frac{\gamma - (\mathcal{P}(B_{i^*}) - \epsilon)}{\gamma} \right) b^* \end{aligned}$$

In other words, the maximum fraction of bids won during exploration and panic is  $\frac{\epsilon + \gamma - \mathcal{P}(B_{i^*})}{\gamma}$ . Note that if the algorithm were to bid  $\mathcal{P}^{-1}(\gamma) \leq p^*$ , then in expectation  $\frac{\gamma - \mathcal{P}(B_{i^*})}{\gamma}$  bids would be  $B_{i^*}$  or above. Therefore, if the

problem is feasible after exploration:

$$\begin{aligned} t &= g(p^*) \\ &\geq g(\mathcal{P}^{-1}(\gamma)) \\ &\geq \frac{\gamma - \mathcal{P}(B_{i^*})}{\gamma} B_{i^*} + \frac{\mathcal{P}(B_{i^*})}{\gamma} g(B_{i^*}) \\ t\phi &\geq \frac{\gamma - \mathcal{P}(B_{i^*})}{\gamma} B_{i^*} \phi + \frac{\mathcal{P}(B_{i^*})}{\gamma} g(B_{i^*}) \\ &\geq \frac{\gamma - \mathcal{P}(B_{i^*})}{\gamma} b^* + \frac{\mathcal{P}(B_{i^*})}{\gamma} g(B_{i^*}) \\ t\phi + \epsilon b^* &\geq \frac{\gamma - \mathcal{P}(B_{i^*})}{\gamma} b^* + \epsilon b^* + \frac{\mathcal{P}(B_{i^*})}{\gamma} g(B_{i^*}) \\ &\geq \frac{\epsilon + \gamma - \mathcal{P}(B_{i^*})}{\gamma} b^* + \frac{\mathcal{P}(B_{i^*})}{\gamma} g(B_{i^*}) \end{aligned}$$

$$fn(t\phi + \epsilon b) \geq C_{\text{explore}} + C_{\text{panic}} + \mathbf{E}[C_{\text{exploit}} | E_1 \dots E_m'] .$$

The last line is due to the fact that  $b^*$  is a bound on the price during exploration and panic,  $g(B_{i^*})$  is a bound on the expected price during exploitation, and the worst-case scenario is to have the maximum number of bids during the exploration and panic phases.  $\square$

LEMMA 19. *If the largest panic bid is  $B_{i^*}$ , during exploitation  $k_s < k_p$  (such that the bid during exploitation is based upon the target spend  $t$ ), the problem is feasible after exploration, and the algorithm has  $\epsilon$ -accurate observations, then  $C \leq tfn + (\epsilon^{1/3} + \epsilon^{2/3})bn$ .*

PROOF SKETCH. This scenario is rare, in that it implies that the algorithm panicked, but then somehow the budget recovered. One possibility is that a very high but unlikely bid was obtained, followed by small but unlikely bids.

Since the last round is  $\epsilon$ -accurate, the argument is similar to Lemma 15.  $\square$

LEMMA 20. *If the largest panic bid is  $B_{i^*}$ , during exploitation  $k_s \geq k_p$  (such that the bid during exploitation is based upon the target number of impressions), the problem is feasible after exploration, and the algorithm has  $\epsilon$ -accurate observations, then  $C \leq \phi tfn + \epsilon nb$ .*

PROOF SKETCH. In this case, we know that  $\mathcal{P}(B_{i^*-1}) \leq \gamma + \epsilon$ , and for however long we are exploring and panicking, we know that  $B_{i^*-1}$  is not expected to give us enough impressions, just as with  $B_{i^*}$  in Lemma 18. If the exploitation bid  $B$  was less than or equal to  $B_{i^*-1}$  (e.g., due to a change in the estimation from  $S_{i^*-1}$  to  $S_{i^*}$ ), then the analysis from Lemma 18 holds, with  $i^* - 1$  replacing  $i^*$ . If  $B > B_{i^*-1}$ , we know that all the bids we obtained during exploration and panicking were not enough, and during the exploitation phase, we choose a bid such that we get just enough impressions above  $B_{i^*-1}$ . Thus, these impressions above  $B_{i^*-1}$  during exploitation and the impressions during exploration and panicking are almost exactly enough such that bidding  $B_{i^*-1}$  during exploitation would have won the remainder. Therefore, an argument similar to Lemma 18 gives a similar result.  $\square$

LEMMA 21. *If the algorithm exits the exploration phase at Line 24, the algorithm has  $\epsilon$ -accurate observations, the problem is feasible after exploration, and  $k_p < k_s$ , then  $C \leq \phi tfn + (\epsilon^{1/3} + \epsilon^{2/3})nb$ .*

PROOF. This is similar to Lemma 17, in that the algorithm does not have more than one impression that is overpriced (note that the last impression won during exploration may be overpriced). Therefore, with the exception that  $i^* = i$ , a variant of Equation 19 holds here.

$$C_{\text{explore}} + C_{\text{panic}} + (n_{\text{exploit}}^*)T_{i^*-1} \leq \phi t f n . \quad (24)$$

If  $B_{\text{final}} \leq B_{i^*-1}$ , then the argument from Lemma 17 holds, in that the algorithm might have well of panicked.

If  $B_{\text{final}} > B_{i^*-1}$ , then the algorithm might as well have panicked: the key condition (that it needs more impressions that it believes will be obtained by  $B_{i^*-1}$ ) is satisfied at the end of the exploration phase, which is loosely what is required in Lemma 20. Note that the bound in Lemma 17 is looser than the bound in Lemma 20.  $\square$

Unfortunately, there are several cases where the algorithm underspends. For instance, suppose that the objective is to obtain a very small number of impressions at a very high price. For instance, 10% of the impressions are at \$11.00, and 90% are at \$1.00, and the target spend is \$2.00, and the algorithm wants 1% of the impressions. Therefore, bidding \$11.00 is ideal. However, if the algorithm starts by bidding \$2.00, it is possible that before the exploration phase is over, the algorithm obtains enough impressions, but has only spent half the budget. Of course, in this case, the algorithm which starts by bidding  $b$  to explore is doing just the right thing.

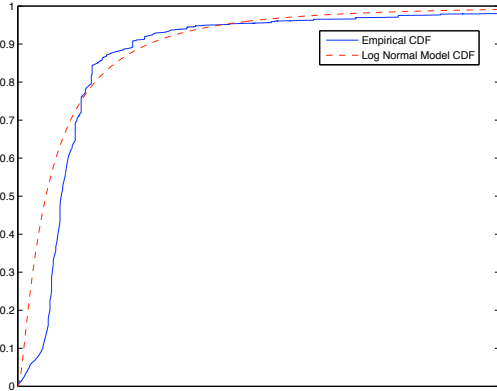


Figure 1: The observed and modeled distribution of the winning bid on the Right Media exchange.

## 4. EXPERIMENTS

In this section we evaluate our algorithms on synthetically derived data drawn from a log-normal distribution, which fits data observed from the Right Media exchange. Algorithms for the fully observable case are evaluated in Section 4.2, and the partially observable case in Section 4.3.

### 4.1 Methodology and Data

To evaluate our algorithms, we collected winning bids from live auctions run on the Right Media exchange, which has over 50,000 buyers and sellers and processes over 6 billion impressions daily. The bids we collected represent a 1% uniform sample across a variety of individual publishers. Focusing on a single publisher, we plot the CDF of the empirical distribution of the bids in Figure 1.

While it is difficult to predict the winning bid on any particular impression, for a fixed publisher the data can be fitted with a log-normal distribution with the appropriate mean and variance. In Figure 1 we provide such a fit for a single publisher (350,000 auctions total). While the exact variance changes from publisher to publisher, on all instances the data can be fitted well with a log-normal distribution.

Based on the above fit, we sample i.i.d. from a log-normal distribution with mean and variance one.<sup>8</sup> While very large prices are never observed on the exchange, the log-normal distribution supports arbitrarily large prices. We deal with this by discarding samples larger than an upper-limit  $b$ , chosen to be the 99.7<sup>th</sup> percentile. That is, we sample from the log-normal distribution conditioned on  $\{X \leq b\}$ .

We set the supply to be  $n = 10,000$  impressions. To evaluate each algorithm we choose 16 evenly spaced values of target  $f$  and target  $t$  from the interval  $(0, 1)$ , and measure the actual fraction and spend against the targets (we include only 4 data points each for clarity). For each of the 256 parameter settings, we run each algorithm on 500 i.i.d. samples to average out effects of sampling.

### 4.2 Results: Fully Observable Exchange

The *Learn-Then-Bid* algorithm waits for a significant exploration period before bidding. An obvious improvement is the *Learn-While-Bid* algorithm which continues to learn during exploitation:  $\mathcal{P}_m, P_m^*, A_m$  and  $Z_m^*$  are updated after *each* bid, as in *Learn-Then-Bid* lines 2–5 but with adaptive targets. Although harder to analyze, this has the advantage of a possibly shorter exploration phase without compromising estimation accuracy. We compare the performance of these two algorithms experimentally: on *average* both algorithms perform very close to ideal. However, *Learn-While-Bid* spending is more *tightly concentrated* around the ideal than *Learn-Then-Bid* for the same exploration phase.

Figures 2 and 3 show the spending of *Learn-Then-Bid* and *Learn-While-Bid* respectively. The dotted lines depict the minimum spending necessary to achieve the target fraction of supply as given by  $\max\{\mathbb{E}[X | X \leq z^*], t\}$ : when the fraction and spend goals cannot simultaneously be satisfied, both algorithms prioritize the former while minimizing over spending. The distribution of each algorithm’s spending, for each  $(f, t)$  pair, is summarized by a box-and-whisker plot: each box depicts the 25%, 50% (the median) and 75% quantiles representing the spread of spending. As is evident in the figures, more runs of *Learn-While-Bid* have spend close to the ideal. To achieve comparable concentration with *Learn-Then-Bid*,  $m$  must be increased. However this can lead to infeasibility for high  $f$  (e.g. for  $m = 1,000$  and  $f = 0.94$  the problem becomes infeasible leading to underdelivery). Figure 4 plots the performance of the algorithms with respect to supply. Both algorithms win close to the target fraction of supply for a wide range of target spending goals.

Figure 5 compares theory and practice, depicting our bounds on the concentration of the fraction of supply won and the spending per impression won, with respect to increasing exploration length. As exploration increases, the bounds become tighter but at the same time the problem becomes

<sup>8</sup>We note that increasing the variance of the log-normal distribution does not affect our algorithms’ behavior; they deliver the same supply and actually overspend less (since there is a larger fraction of lower-priced impressions).



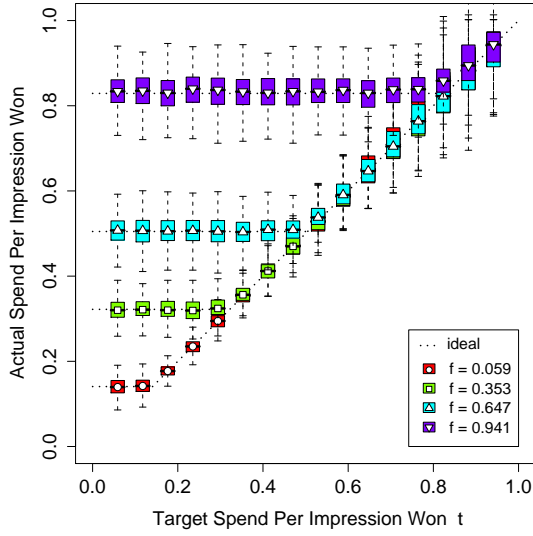


Figure 2: Actual & ideal spend per impression won by *Learn-Then-Bid* with  $m = 100$  for 4 values of  $f$ .

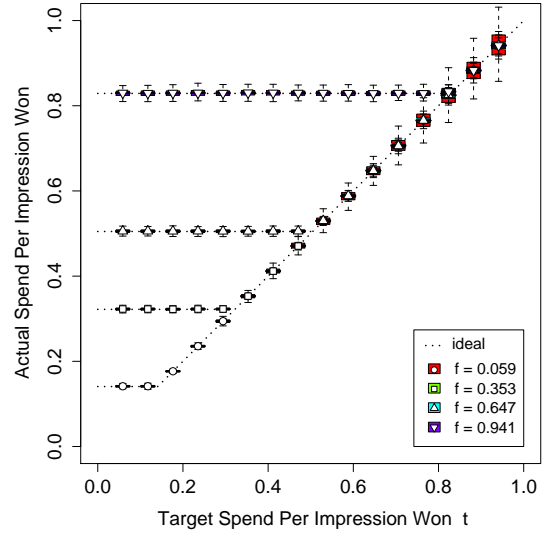


Figure 3: Actual & ideal spend per impression won by *Learn-While-Bid* with  $m = 100$  for 4 values of  $f$ .

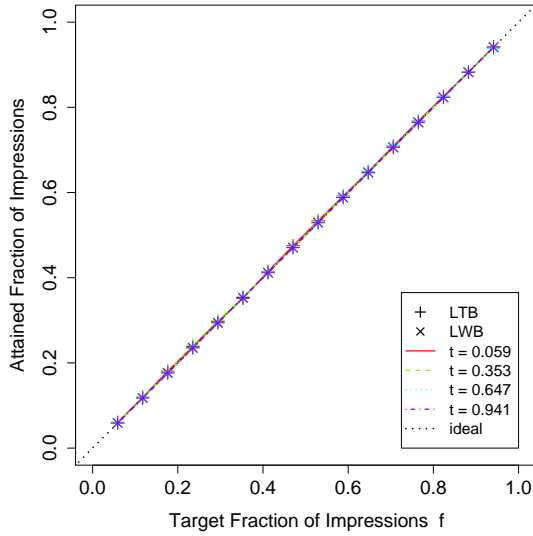


Figure 4: Fraction of supply won by *Learn-Then-Bid* and *Learn-While-Bid* with  $m = 100$  for 4 values of  $t$ .

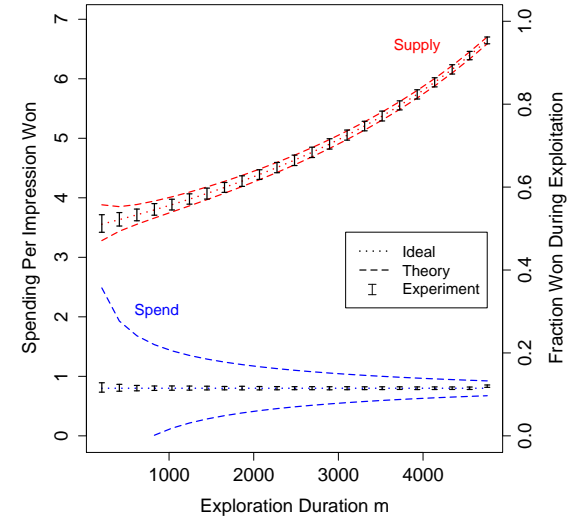


Figure 5: *Learn-Then-Bid* supply and spend 95% confidence bands given by: the concentration bounds, and 1,000-sample empirical quantiles.

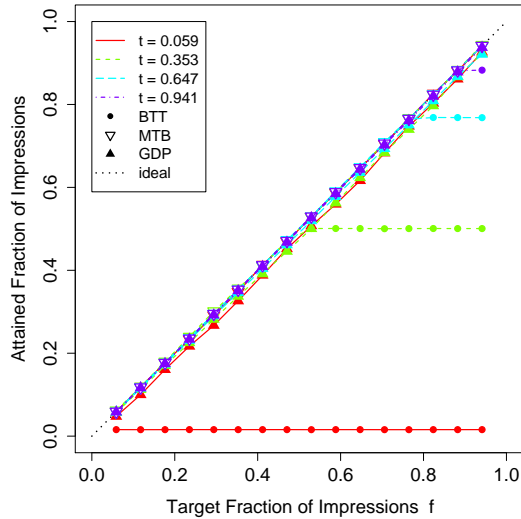
harder to satisfy until it becomes infeasible (occurring at the maximum  $m$  shown). Also shown are empirical results for running *Learn-Then-Bid*. In particular for each  $m$  value (exploration phase length), *Learn-Then-Bid* was run 1,000 times. For each run the fraction won during exploitation, and the spend per impression won, were calculated and the 95% two-sided quantiles were computed and overlaid. In this case the distribution-free theory matches the empirical results well, particularly for concentration of fraction won.

### 4.3 Results: Partially Observable Exchange

In this section we evaluate algorithms for bidding in a partially observable exchange. The *Guess-Double-Panic* al-

gorithm outperforms the strawman bidders in terms of target spend for a wide range of target supply fractions. More interestingly, the lack of full information does not handicap the *Guess-Double-Panic* algorithm: the accuracy of supply and spend are very close to those for full information.

Figures 6 and 7 plot the supply and spend for *Guess-Double-Panic* as well as two strawman algorithms, *Max-Then-Bid* and *Bid-Two-t*. Given that *Learn-Then-Bid* performs well even for very short exploration phases, it is natural to apply the same idea in the partially observable setting: *Max-Then-Bid* bids  $b$  in the length  $m$  exploration phase and



**Figure 6:** Fraction of supply won by *Bid-Two-t*, *Max-Then-Bid* and *Guess-Double-Panic* with  $m = 100$  for 4 values of  $t$ .

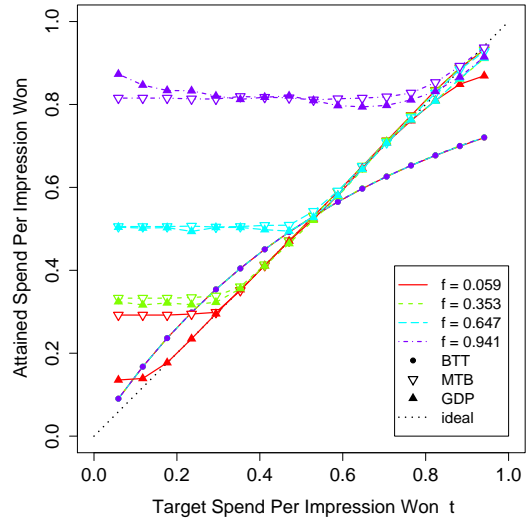
then exploits based on the empirical distribution.<sup>9</sup>

While *Max-Then-Bid* performs well on supply it is an inferior strategy for target spend, since the spending during exploration can be very high: specifically when  $f$  and  $t$  are both small the algorithm must acquire the right number of impressions at a low price, and *Max-Then-Bid* is very likely to overspend. *Guess-Double-Panic* will perform well precisely in this setting due to its more cautious exploration starting with a bid of  $t$  as opposed to  $b$ . This comparison is borne out in Figure 7 where *Max-Then-Bid* overspends most dramatically relative to *Guess-Double-Panic* for the lowest target fraction  $f = 0.059$ ; overspending is evident for higher values up to  $f = 0.59$ . The *Bid-Two-t* strawman algorithm does not perform much better than expected from theory, which predicts supply and spend within only a loose multiplicative factor of the targets.

## 5. CONCLUSIONS

We study the problem of acquiring a given number of impressions with a given budget constraint by bidding against an *unknown* distribution. Our approach consists of learning the distribution in an exploration phase, and then bidding according to the empirical distribution from observations during exploration. We consider both the fully observable and the harder partially observable case, and present algorithms with theoretical performance guarantees that also perform very well in experimental evaluations against realistic data. The experiments indicate that in addition to performing well with respect to both constraints, our algorithm for partial information does nearly as well as algo-

<sup>9</sup>*Max-Then-Bid* dominates *Learn-Then-Bid* in number of impressions acquired despite partial information: *Max-Then-Bid* wins every impression during exploration in contrast to *Learn-Then-Bid* which wins zero. Also, if the problem is feasible after *Learn-Then-Bid*'s exploration then it will be feasible for *Max-Then-Bid*.



**Figure 7:** Spend per impression won by *Bid-Two-t*, *Max-Then-Bid* and *Guess-Double-Panic* with  $m = 100$  for 4 values of  $f$ .

rithms in the full information setting despite the fact that a cost must be paid for every sample during exploration. The performance of the algorithms improves as the supply increases, and is asymptotically optimal since a longer exploration phase can be used for higher accuracy; also, the actual number of impressions won and the total spend are more tightly concentrated around their means as  $n$  increases.

The most interesting direction for further research is removing the iid assumption, and considering a game theoretic perspective: we assume that every other bidder in each individual auction has unit demand and therefore bids his value; one can instead consider best response and equilibrium analysis when a number of bidding agents compete in such a marketplace.

## Acknowledgements

We gratefully acknowledge the support of the NSF through grant DMS-0707060.

## References

- [1] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47:235–256, 2002.
- [2] M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, A. Karlin, C. Mathieu, and M. Schwarz. Greedy bidding strategies for keyword auctions. In *ACM EC*, 2007.
- [3] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *Annals of Mathematical Statistics*, 27(3):642–669, 1956.
- [4] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, February 1994.
- [5] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.