

# Asymptotically Optimal Algorithm for Stochastic Adwords

Nikhil R. Devanur, Microsoft Research, Redmond. [nikdev@microsoft.com](mailto:nikdev@microsoft.com).

Balasubramanian Sivan, Computer Sciences Dept., University of Wisconsin-Madison.  
[balu2901@cs.wisc.edu](mailto:balu2901@cs.wisc.edu).

Yossi Azar, School of Computer Science, Tel-Aviv University. [azar@tau.ac.il](mailto:azar@tau.ac.il).

In this paper we consider the adwords problem in the *unknown distribution* model. We consider the case where the budget to bid ratio  $k$  is at least 2, and give improved competitive ratios. Earlier results had competitive ratios better than  $1 - 1/e$  only for “large enough”  $k$ , while our competitive ratio increases continuously with  $k$ . For  $k = 2$  the competitive ratio we get is 0.729 and it is 0.9 for  $k = 16$ . We also improve the asymptotic competitive ratio for large  $k$  from  $1 - O(\sqrt{\log n/k})$  to  $1 - O(\sqrt{1/k})$ , thus removing any dependence on  $n$ , the number of advertisers. This ratio is optimal, even with known distributions. That is, even if an algorithm is tailored to the distribution, it cannot get a competitive ratio of  $1 - o(\sqrt{1/k})$ , whereas our algorithm does not depend on the distribution. The algorithm is rather simple, it computes a *score* for every advertiser based on his original budget, the remaining budget and the remaining number of steps in the algorithm and assigns a query to the advertiser with the highest bid plus his score. The analysis is based on a “hybrid argument” that considers algorithms that are part actual, part hypothetical, to prove that our (actual) algorithm is better than a completely hypothetical algorithm whose performance is easy to analyze.

Categories and Subject Descriptors: J.4 [Social and Behavioral Sciences]: Economics; F.2.0 [Analysis of Algorithms and Problem Complexity]: General

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Adwords, online algorithms, stochastic setting

## 1. INTRODUCTION

Consider the following problem faced by a search engine: a sequence of queries arrive online and have to be allocated immediately upon arrival to one of several competing advertisers who are interested in the query. Different advertisers could have different bids for a given query, and the search engine needs to determine which advertiser to choose, and how much payment to charge him. This has been a source of many interesting problems, and there have been a large number of papers on it, modeling various aspects of this problem, and abstracting out certain characteristics while ignoring others. One such formalization that captures the online algorithmic aspect is the “adwords” problem introduced by Mehta et al. [Mehta et al. 2005]. In this model, a query is characterized by the vector of revenues obtained if matched to an advertiser (called bids). When allocated a query, an advertiser simply pays his bid for that query (ignoring the game theoretic aspects that go into advertiser bidding). The problem asks for the online query allocation algorithm that maximizes the search engine’s revenue, subject to the constraint that the revenue from any given advertiser is capped by his budget.

---

Part of this work was done while B. Sivan and Y. Azar were visiting Microsoft Research, Redmond, WA. The work of Y. Azar was supported in part by the Israel Science Foundation (grant No. 1404/10).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

EC’12, June 4–8, 2012, Valencia, Spain.

Copyright 2012 ACM 978-1-4503-1415-2/12/06...\$10.00.

We consider the adwords problem in the *stochastic unknown distribution model*. In this model, each query in the sequence is sampled independently from an underlying distribution but the algorithm does not know this distribution. In other words the algorithm is independent of the distribution but its performance is measured in expectation over the draws from the distribution.

The performance of an algorithm is measured by the competitive ratio: the ratio of the revenue of the algorithm to the optimal revenue<sup>1</sup> on hindsight. The competitive ratio for the adwords problem typically depends on a key parameter, called the smallest budget to bid ratio, denoted here by  $k$ . [Devanur et al. 2011] gave near tight bounds for the adwords problem (and for a more general class of problems called resource allocation problems) in the stochastic unknown distribution model. If  $k = 1$  then<sup>2</sup> the competitive ratio is  $1 - 1/e$ , or if  $k = \log n/\epsilon^2$  for some  $\epsilon > 0$  and  $n$  is the number of advertisers, then the competitive ratio is  $1 - O(\epsilon)$ .

The second ratio is better than the first only for  $k$  large enough. This left a gap where for a large range of  $k$  the best competitive ratio was still  $1 - 1/e$ . This raises the question, what can we say if  $k = 2$ ? Or  $k = 3$ ? Is there an algorithm for which the competitive ratio gradually increases with  $k$ ? This was left as an open question in [Devanur et al. 2011].

We answer this question in the affirmative in this paper. We give an algorithm with a competitive ratio of  $1 - \frac{k^k}{k!e^k} \geq 1 - \frac{1}{\sqrt{2\pi k}}$ . This gives significant improvements right away, for instance for  $k = 2$  it is 0.729, and it is already 0.9 for  $k = 16$ . The competitive ratio for a range of small  $k$  is summarized in Table I.

Table I. Competitive ratio for small  $k$ .

$k$	$1 - \frac{k^k}{k!e^k}$
1	0.63
2	0.73
3	0.77
10	0.87
20	0.91
50	0.94
100	0.96

Somewhat surprisingly, this result also gives an improvement over [Devanur et al. 2011] for large  $k$ . In particular,  $k$  only needs to be larger than  $1/\epsilon^2$  in order to get a  $1 - O(\epsilon)$  competitive ratio, getting rid of a factor of  $\log n$ , and more importantly, any dependence on  $n$ . This competitive ratio is *optimal*, up to constant factors; no algorithm can get a competitive ratio of  $1 - o(1/\sqrt{k})$ . In fact this upper bound even applies to algorithms that *know* the distribution. In other words even if an algorithm could be tailored to the underlying distribution, one cannot get a better competitive ratio.

The algorithm we present addresses another unpleasant nature common in previous algorithms for this problem. The competitive ratio of the earlier algorithms typically depends on the *worst* budget to bid ratio. That is, the budget to bid ratio is defined for each advertiser and the competitive ratio depends on the smallest of these. This means that introducing even one advertiser with a small budget can completely destroy the competitive ratio guaranteed by the analysis. Our algorithm does not suffer from this

<sup>1</sup>Some care is needed to define this in the stochastic model; see Section 2 for a precise definition.

<sup>2</sup>The budget is always greater than any bid w.l.o.g., so  $k \geq 1$ .

and the guarantee given by the analysis gracefully degrades with such changes. The details are presented in Section 2.

These improvements in the competitive ratio come at a price, though. Our guarantees are only “in expectation” whereas the ones in [Devanur et al. 2011] were “with high probability”. This seems unavoidable though. Another point to note is that the algorithms in [Devanur et al. 2011] are applicable to more general “resource allocation” problems, whereas our algorithm does not generalize. This is just as well since for these general resource allocation problems, it is unlikely that the factor of  $\log n$  can be avoided. Finally, we have to assume that in the corresponding *distribution instance*<sup>3</sup>, either all the budgets are saturated, or, the algorithm knows the consumption of the budgets in an optimal solution to the distribution instance, whereas the algorithms in [Devanur et al. 2011] do not depend on any distributional parameters at all. We note here that knowing the optimal consumption is information theoretically a strictly weaker requirement than knowing the entire distribution. Knowing the optimal consumption  $C_i$  for advertiser  $i = 1 \dots n$ , requires knowledge of  $n$  parameters, as against the knowledge of the entire distribution, which could have a very large (even infinite) support.

On the other hand our algorithm also works when the distribution is changing over time, in particular, in the Adversarial Stochastic Input (ASI) model (introduced by [Devanur et al. 2011]), just like the algorithms in [Devanur et al. 2011]. In fact, under the assumption that the budgets are saturated, this model simply allows a different distribution to be picked in each step. (See Section 6 for a precise definition.)

In summary, this algorithm almost closes the last few gaps left open for the adwords problem in the stochastic unknown distribution model.

### 1.1. Related Work

As mentioned earlier, in the stochastic unknown distribution model [Devanur et al. 2011] gave a  $1 - O(\epsilon)$  competitive algorithm for  $k = \log n/\epsilon^2$ , improving upon earlier work by [Devanur and Hayes 2009; Charles et al. 2010].

There has been a lot of work done on the special case of online bipartite matching. This is a special case of adwords where all the budgets are 1 and the bids are either 0 or 1. In this case, the budget to bid ratio,  $k$ , is 1. Most of the results have focused on showing that the bound of  $1 - 1/e$  can be beaten in the *known* distribution model [Feldman et al. 2009; Bahmani and Kapralov 2010; Manshadi et al. 2011]. The best competitive ratio achieved in this line of work is 0.7036 [Haeupler et al. 2011]. These results are incomparable to ours: they relax the stochastic model by allowing the algorithm to be tailored to the underlying distribution whereas we relax the budget to bid ratio. We also consider the more general adwords problem. We show that allowing even slightly larger budget to bid ratios can give a significantly better competitive ratio. Since the main motivation for these problems originate in advertising, this seems like a natural relaxation.

More recently [Karande et al. 2011; Mahdian and Yan 2011] consider the online bipartite matching problem in the unknown distribution model. The best competitive ratio in this series is 0.69 [Mahdian and Yan 2011]. While the stochastic model in these papers coincides with ours, we consider the more general weighted version and the range  $k > 1$ .

The competitive ratio in the worst case model with  $k = 1$  is still only  $1/2$ , while with large  $k$  the competitive ratio tends to  $1 - 1/e$  [Mehta et al. 2005; Buchbinder et al. 2007]. Moreover  $1 - 1/e$  is optimal, so the interesting range in the worst case is  $[1/2, 1 - 1/e]$ .

---

<sup>3</sup>See Section 2.2 for a definition.

## 1.2. The Algorithm

The main idea behind the algorithm is as follows. In each step we would like to match the query to the advertiser that maximizes the sum of the revenue obtained in this step and the *residual* revenue that can be obtained in the rest. The problem of course is that the naïve way to estimate the residual revenue requires knowledge of the distribution. The key insight is that we can however estimate this residual revenue in a *distribution independent* way, knowing only the consumption of the advertisers in an optimal solution to the distribution instance.

The “magical” estimate comes from the analysis of a hypothetical algorithm that knows the distribution. It boils down to analyzing the following simple balls and bins process: in every step a ball is thrown into one of  $n$  bins with certain capacities and according to some probability distribution. If this is repeated for  $m$  steps then what is the expected number of balls that “overflow”?

In the rest of this section, we show an almost complete proof for a special case of the problem. This already contains some of the main ideas used for the more general results. Consider a bipartite graph  $G = (L, R, E)$ , with  $|L| = n$  and  $|R| = m$ . Suppose we repeat  $m$  times the following procedure  $\mathcal{P}$ : a vertex is chosen uniformly at random from  $R$  (with replacement) and is matched to one of its neighbors in  $L$ . For each vertex  $i$  in  $L$  the number of times it can be matched is at most  $B_i$ .

Now suppose that the matching has to be done without knowing the bipartite graph. In each step we only see the neighbors of the vertex in  $R$  chosen at that step. The  $B_i$ 's are known in advance. We wish to maximize the number of matches. Call such a procedure  $\mathcal{P}^u$ .

Suppose that  $G$  has a perfect matching, that is every vertex  $j$  in  $R$  can be matched to one of its neighbors, say  $M(j)$  in  $L$  such that every vertex  $i$  in  $L$  is matched exactly  $B_i$  times. Consider a version of procedure  $\mathcal{P}$  that does the following: if in a given step the vertex chosen is  $j$  then it is matched to  $M(j)$ . If  $i = M(j)$  is already matched  $B_i$  times, then it is discarded. Call this procedure  $\mathcal{P}^k$ . In this procedure, for a given  $i \in L$  the probability that it is matched in a given step is exactly  $B_i/m$ . Therefore the expected number of times  $i$  is matched throughout the entire procedure is at least  $B_i - \sqrt{B_i/2\pi}$ . (See Lemma 3.1.)

We now define procedure  $\mathcal{P}^u$  inductively. Suppose that we have defined  $\mathcal{P}^u$  up to step  $t-1$ . Consider step  $t$ . Now consider a *hybrid* procedure  $\mathcal{H}_t$  that runs  $\mathcal{P}^u$  for  $t$  steps and  $\mathcal{P}^k$  for the remaining steps. Step  $t$  of  $\mathcal{P}^u$  is chosen so that the expected number of matches in  $\mathcal{H}_t$  is maximized. This can be done without knowing  $G$  since the expected number of matches for a vertex  $i$  in  $L$  in the remaining steps of  $\mathcal{P}^k$  only depends on the number of times  $i$  is already matched, the probability of a match in one step,  $B_i/m$ , and the number of remaining steps,  $m-t$ . (See Section 3.2.1)

Now consider the expected number of matches in  $\mathcal{H}_t$  and  $\mathcal{H}_{t-1}$ . These two differ only in step  $t$ . The choice of the vertex to match to in step  $t$  in  $\mathcal{H}_t$  was defined to maximize the expected number of matches in the remaining steps. However, for  $\mathcal{H}_{t-1}$  this was simply a fixed choice given by  $M$ . Hence the expected number of matches is only higher in  $\mathcal{H}_t$ .

It is easy to see that  $\mathcal{H}_0$  is identical to  $\mathcal{P}^k$  and  $\mathcal{H}_m$  is identical to  $\mathcal{P}^u$ . Thus the expected number of matches in  $\mathcal{P}^u$  is only higher than that in  $\mathcal{P}^k$ . For the latter, this number is at least  $\sum_i B_i - \sqrt{B_i/2\pi}$ .

## 2. PRELIMINARIES AND MAIN RESULTS

### 2.1. Adwords problem

In the adwords problem, there are  $n$  advertisers, with advertiser  $i$  having a budget of  $B_i$ . There are  $m$  queries that arrive in a sequence. When query  $j$  arrives, the bid  $b_{ij}$

of each advertiser  $i$  is revealed to the algorithm. Upon assigning query  $j$  to advertiser  $i$ , the algorithm gets a revenue of  $b_{ij}$ , and the budget of advertiser goes down by  $b_{ij}$ . The algorithm is allowed to assign a query to an advertiser even if his remaining budget at that point is strictly less than his bid for the query, but get a revenue of just the remaining budget and not the full bid. Each query can be assigned to at most one advertiser, a query has to be assigned without knowing which queries arrive in the future, and an assignment once made cannot be changed. The objective is to maximize the total revenue subject to the budget constraints. For a given fixed sequence of queries  $1, 2, \dots, m$ , the optimal objective value of the following LP is an upper bound on the maximum revenue obtainable.

$$\begin{aligned}
& \text{maximize } \sum_j b_{ij} \cdot x_{ij} \text{ s.t.} & (1) \\
& \forall i, \sum_j b_{ij} \cdot x_{ij} \leq B_i \\
& \forall j, \sum_i x_{ij} \leq 1. \\
& \forall i, j, x_{ij} \geq 0
\end{aligned}$$

Since queries arrive online, the LP gets revealed one column at a time. The optimum solution to the full LP only depends on the *set* of queries and not on the order of arrival.

Let  $\gamma_i = \max_j \frac{b_{ij}}{B_i}$  be the maximum bid-to-budget ratio for advertiser  $i$ , and let  $\gamma = \max_i \gamma_i$  be the global largest bid-to-budget ratio. The algorithm knows  $m$  and  $\gamma_i$ 's. Note that knowing  $\gamma_i$ 's is same as knowing the maximum bid  $b_i = \max_j b_{ij}$  of every advertiser  $i$ .

## 2.2. Distribution instance

We consider the unknown distribution model for the adwords problem. In this model each query is drawn i.i.d from a distribution unknown to the algorithm.

The benchmark for all the competitive ratios in this paper is an upper bound on the expected optimal (fractional) solution to LP 1, where the expectation is over the distribution from which queries are drawn, i.e., expectation over all possible sequences of  $m$  queries. We emphasize that the optimal solution refers to the solution obtained after knowing the bid values of all the queries that have arrived, i.e., after the full LP has been revealed. In order to describe the benchmark, we first describe the distribution instance — an offline instance which is a function of the distribution. Every query in the support of the distribution is a query in this instance. The budgets of the advertisers in this instance are the same as in the original instance. The bid of advertiser  $i$  for query  $j$  is given by  $mp_j b_{ij}$ , where  $p_j$  is the probability with which query  $j$  is drawn. The intuition is that if queries were drawn from a distribution, then the expected number of times query  $j$  will be seen is exactly  $mp_j$ , and this is reflected in the distribution instance by scaling the bids. In sum, the distribution instance is the following:

$$\begin{aligned}
& \text{maximize} && \sum_{j \text{ in the support}} mp_j b_{ij} \cdot x_{ij} \text{ s.t.} && (2) \\
& \forall i, \sum_j mp_j b_{ij} \cdot x_{ij} \leq B_i \\
& \forall j, \sum_i x_{ij} \leq 1. \\
& \forall i, j, x_{ij} \geq 0
\end{aligned}$$

It turns out that the fractional optimal solution to the distribution instance is an upper bound on the expectation of OPT, where OPT is the offline fractional optimum of the actual sequence of queries.

LEMMA 2.1. ([Devanur et al. 2011])  $\text{OPT}[\text{Distribution instance}] \geq \mathbb{E}[\text{OPT}]$

The *competitive ratio* of an algorithm is defined as the ratio of the expected profit of the algorithm to the fractional optimal profit for the distribution instance.

### 2.3. Results

**2.3.1. Main result.** Our main result for the adwords problem is with an extra assumption that the algorithm is given the amounts of budget consumed (for each advertiser) by some optimal solution to the distribution instance. As discussed in the introduction, this requirement is information theoretically strictly weaker than requiring to know the entire distribution. We prove that given this, there is a simple online algorithm that achieves the asymptotically optimal competitive ratio of  $1 - O(\sqrt{\gamma})$ . As a corollary we also get a result without this extra assumption in the case where the optimal solution saturates all the budgets.

THEOREM 2.2. *Given the budget consumption  $C_i$  from every advertiser  $i$  by some optimal solution to the distribution instance, the adwords problem has an online algorithm that achieves a revenue of  $\sum_i C_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$ .*

Remark 2.3. We note the following about Theorem 2.2

- (1) Often it could be the case that many advertisers have small  $\gamma_i$ 's but a few advertisers have large  $\gamma_i$ 's, thus driving the maximum  $\gamma$  up. But our algorithm's competitive ratio doesn't degrade solely based on the maximum  $\gamma$ . Rather, it depends on all  $\gamma_i$ 's and is robust to a few outliers.
- (2) Our algorithm performs slightly better than what is stated. Let  $k_i = \frac{1}{\gamma_i}$ . Then our algorithm attains a revenue of  $\sum_i C_i(1 - \frac{k_i^{k_i}}{k_i!e^{k_i}}) \geq \sum_i C_i(1 - \frac{1}{\sqrt{2\pi k_i}}) = \sum_i C_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$ .

An immediate corollary of Theorem 2.2 is the following. Note that  $\gamma = \max_i \gamma_i$ .

COROLLARY 2.4. *Given the consumption information for every advertiser  $i$ , the adwords problem has an online algorithm with a competitive ratio of  $1 - O(\sqrt{\gamma})$ .*

Remark 2.5. We note the following about Corollary 2.4.

- (1) The competitive ratio of  $1 - O(\sqrt{\gamma})$  is asymptotically optimal. A simple example proving this claim is shown in Section 3.3.
- (2) This corollary also improves on the best competitive ratio possible in the stochastic setting from  $1 - O(\sqrt{\gamma \log n})$  [Devanur et al. 2011] to  $1 - O(\sqrt{\gamma})$ .

(3) Another way of stating Corollary 2.4 is that when  $\gamma \leq \epsilon^2$ , there is an online algorithm that achieves a  $1 - O(\epsilon)$  competitive ratio.

Another corollary of Theorem 2.2 is its application to the special case of online stochastic matching problem introduced by [Feldman et al. 2009]. In this problem, all the budgets are equal to 1, and all the bids are either 0 or 1. [Feldman et al. 2009] showed a competitive ratio of 0.67, and this has been improved to 0.702 by [Manshadi et al. 2011]. The corollary is that if the budgets are increased to 2 from 1, the competitive ratio can be improved to 0.729, assuming that the algorithm is given the number of matches to every left-hand-side (LHS) vertex. It increases even further for the  $B$ -matching problem, to a competitive ratio of  $1 - \frac{1}{\sqrt{2\pi B}}$ . More generally for the  $B$ -matching problem, where LHS vertex  $i$  can accept  $B_i$  matches, the matching size obtained is  $\sum_i C_i (1 - \frac{1}{\sqrt{2\pi B_i}})$  thus giving a factor of at least  $1 - \frac{1}{\sqrt{2\pi B}}$  where  $B = \min_i B_i$ .

**COROLLARY 2.6.** *For the online stochastic  $B$ -matching problem, there is an online algorithm that achieves a revenue of  $\sum_i C_i (1 - \frac{1}{\sqrt{2\pi B_i}})$ , and thus a competitive ratio of at least  $1 - \frac{1}{\sqrt{2\pi B}}$ , where  $B = \min_i B_i$ , provided the algorithm is given the number of matches  $C_i$  to every LHS vertex in some optimal solution to the distribution instance.*

Note that this shows an interesting trade-off w.r.t. the results of [Feldman et al. 2009]. There is a big improvement possible by just letting the number of possible matches in the LHS to 2 instead of one, and this is evidently the case for most applications of matching motivated by online advertising. On the other hand, instead of having to know the distribution it is enough to know the optimal expected consumptions.

**2.3.2. Saturated instances.** We call an adwords instance a saturated instance whenever there exists some optimal solution to the distribution instance such that all advertisers' budgets are fully consumed, i.e.,  $C_i = B_i$ . Since  $B_i$ 's are known to the algorithm anyway, this means that for saturated instances the dependence on any distributional parameter is removed. That is, Theorem 2.2 when specialized to a saturated instance, will give us an algorithm that achieves a revenue of  $\sum_i B_i (1 - \sqrt{\frac{\gamma_i}{2\pi}})$  without the knowledge of any distributional parameter.

**THEOREM 2.7.** *For all  $\gamma$ , any saturated instance of the adwords problem in the unknown distributional model has an online algorithm that achieves a revenue of  $\sum_i B_i (1 - \sqrt{\frac{\gamma_i}{2\pi}})$  and thus a competitive ratio of at least  $1 - O(\sqrt{\gamma})$ .*

On similar lines, Corollaries 2.4 and 2.6 can be specialized to saturated instances, and hence we can remove in them the knowledge of budget consumption assumption. We don't state them here.

**2.3.3. Unweighted instances.** The most fulfilling result would be to eliminate dependence on any distributional parameter at all, not just for saturated instances (as we did in Section 2.3.2) but for general instances. We make some progress toward this goal for a special case of the adwords problem. We show that when every advertiser's bid is either  $b$  or zero, there is an online algorithm that achieves a competitive ratio of  $1 - O(\sqrt{\gamma} \frac{\sum_i B_i}{\text{OPT}})$ , where,  $\text{OPT} = \sum_i C_i < \sum_i B_i$  is the total consumption. This is an improvement over the factor of  $1 - O(\gamma^{1/7} \frac{\sum_i B_i}{\text{OPT}})$  given by [Mirrokni et al. 2012] using the Balance algorithm of [Kalyanasundaram and Pruhs 1998]. Besides, their definition of  $\gamma$  is always at least as large as ours, and in some cases, could be strictly larger. On the other hand, their result holds for random permutations, whereas our proof is for i.i.d with unknown distributions.

We state the result below formally. We call an instance an unweighted instance when the bids are all of the form 1 or 0. Any result for such instances automatically carry forward to instances where the bids are of the form  $b$  or 0.

**THEOREM 2.8.** *The unweighted instance of the adwords problem in the unknown distributional model has an online algorithm that achieves a competitive ratio of  $1 - \sqrt{\frac{\pi}{2\pi} \frac{\sum_i B_i}{\text{OPT}}}$ .*

*Organization.* Section 3 deals with saturated instances and proves Theorem 2.7. We use this result in Section 4 that deals with general instances where the algorithm is given the budget consumption of every advertiser in some optimal solution to the distribution instance, and proves Theorem 2.2. Section 5 deals with the special case of unweighted instances with no knowledge of distribution and proves Theorem 2.8.

### 3. SATURATED INSTANCES

In this section we describe the algorithm for saturated instances of the adwords problem.

#### 3.1. Hypothetical-Oblivious algorithm

Before presenting our algorithm, we describe a hypothetical algorithm called Hypothetical-Oblivious that also achieves a revenue of  $\sum_i B_i (1 - \sqrt{\frac{\gamma_i}{2\pi}})$ . The Hypothetical-Oblivious algorithm uses the solution to the distribution instance (which is an offline instance) to perform the online allocation of queries. Note that since the distribution is unknown to the algorithm designer, the distribution instance cannot be computed, and thus Hypothetical-Oblivious is a hypothetical algorithm. Let  $\{x_{ij}\}$  denote some optimal solution to the distribution instance. When query  $j$  arrives, the algorithm Hypothetical-Oblivious assigns it to advertiser  $i$  with probability  $x_{ij}$ . Thus Hypothetical-Oblivious is a non-adaptive algorithm that follows the same assignment probabilities irrespective of the previously arrived queries and their assignments. Even if the budget of an advertiser has been fully consumed, Hypothetical-Oblivious does not make any alterations to the said assignment rule, i.e., it will get zero revenue of such allocations.

**3.1.1. Hypothetical-Oblivious algorithm on single-bid instances.** In this section, we restrict attention to instances where advertiser  $i$ 's bids are either  $b_i$  or zero, and relax this assumption in Section 3.1.2. The Hypothetical-Oblivious algorithm, at any given time-step, assigns a query to advertiser  $i$  with probability  $\sum_j p_j b_{ij} x_{ij} = B_i/m$ , where  $b_{ij} \in \{0, b_i\}$ , and the equality follows from the fact that the solution  $\{x_{ij}\}$  consumes the entire budget in the distribution instance. When bids are 0 or  $b_i$ , observe that the Hypothetical-Oblivious algorithm is simply a balls and bins process which, in each time-step throws a ball into bin  $i$  with probability  $\frac{B_i}{b_i m}$ . Note that full consumption implies that  $m \geq \sum_i \frac{B_i}{b_i}$ . We now show that the expected number of balls(queries) in bin(advertiser)  $i$  at the end of this process is at least  $\frac{B_i}{b_i} \left(1 - \sqrt{\frac{b_i}{2\pi B_i}}\right) = \frac{B_i}{b_i} \left(1 - \sqrt{\frac{\gamma_i}{2\pi}}\right)$ . Since each ball is worth  $b_i$ , this proves that the revenue in bin  $i$  at the end is  $B_i \left(1 - \sqrt{\frac{\gamma_i}{2\pi}}\right)$ .

**LEMMA 3.1.** *In a balls and bins process where a given bin with capacity  $k$  receives a ball with probability  $k/m$  at each step, the expected number of balls in the given bin after  $m$  steps is at least  $k \left(1 - \frac{1}{\sqrt{2\pi k}}\right)$ .*

**PROOF.** Let  $X_m$  be the binomial random variable denoting the number of balls thrown at the bin, when  $m$  is the total number of balls, and  $k/m$  is the probability



with which a ball is thrown at the bin. Clearly  $E[X_m] = k$ . The quantity we are interested in, that is the expected number of balls in the bin after  $m$  steps, is  $E[\min(X_m, k)]$ . This quantity, as proved in [Yan 2011], monotonically decreases in  $m$ . In other words, more balls get wasted (due to overflow) if we throw  $m + 1$  balls with probability  $\frac{k}{m+1}$  each, instead of  $m$  balls with probability  $\frac{k}{m}$  each. Therefore the quantity  $E[\min(X_m, k)]$  attains its minimum as  $m \rightarrow \infty$ , and equals  $k(1 - \frac{k^k}{k!e^k}) \geq k(1 - \frac{1}{\sqrt{2\pi k}})$  [Yan 2011].  $\square$

Thus the competitive ratio achieved by Hypothetical-Oblivious is at least  $1 - \frac{k^k}{k!e^k}$  where  $k = \lfloor 1/\gamma \rfloor$ .

**COROLLARY 3.2.** *For any single-bid instance of the adwords problem that is saturated, the Hypothetical-Oblivious algorithm achieves a revenue of  $\sum_i B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$ .*

**3.1.2. Extending Hypothetical-Oblivious to arbitrary bids.** We now show that the Hypothetical-Oblivious algorithm can actually achieve the same revenue of  $\sum_i B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$  for arbitrary bids.

In the arbitrary bids case, advertiser  $i$ 's bids are in  $[0, b_i]$  (instead of  $\{0, b_i\}$  of the previous section). That is,  $b_i$  is just the maximum bid and several other smaller bids are also possible. The Hypothetical-Oblivious algorithm for such instances is like a balls and bins process albeit the balls can be fractional balls. That is in each step, a ball of size  $s \in [0, b_i]$  is thrown into the bin  $i$ , where  $s = b_{ij}$  with probability  $p_j x_{ij}$ , and thus, the expected "amount" of ball aimed at bin  $i$  in a single step is  $\sum_j p_j b_{ij} x_{ij} = B_i/m$ . Our argument is that for every bin, any arbitrary bid instance (which we also refer to as fractional bid instance) never performs worse in expectation compared to the corresponding single-bid instance (which we also refer to as integral bid instance). Thus, from now on we fix some particular bin, say  $i$ , and the random variables we define in the rest of this section 3.1.2 are with respect to this particular bin  $i$ , though we drop this index.

Let the random variables  $X_F^j$  and  $X_I^j$  denote respectively, the amount of ball aimed at a given bin in step  $j$ , when the bids are in  $[0, b]$  and  $\{0, b\}$ . Since budgets are fully consumed, we have that for a given bin of capacity  $k$ , the expectations of  $X_F^j$  and  $X_I^j$  are equal, i.e.,  $E[X_F^j] = E[X_I^j] = k/m$ .

Let the random variables  $X_F$  and  $X_I$  denote respectively, the total amount of balls aimed at the given bin over all the steps, in the fractional bid and the integral bid case. That is,  $X_F = \sum_j X_F^j$ , and  $X_I = \sum_j X_I^j$ . The amount of ball that has landed in the given bin (in the fractional bid case) is given by  $\min(k, X_F)$ , and thus  $E[\min(k, X_F)]$  is the quantity we are interested in. Similarly  $E[\min(k, X_I)]$  is the quantity of interest for integral bid case. By Lemma 3.1, we know that in the integral bid case the expected number of balls landed in the bin is  $E[\min(k, X_I)] \geq k(1 - \frac{k^k}{k!e^k})$ . If we prove this inequality for  $X_F$  also, that completes the proof. For a given expected amount of ball in the balls and bins process (in this case we have  $k/m$  amount of ball thrown in expectation in each step), the maximum wastage of balls (due to overflow) occurs when the distribution of ball size is extreme, i.e., either  $b$  or zero. Thus, the wastage for the  $[0, b]$  case is at most the wastage for  $\{0, b\}$  case, and hence  $E[\min(k, X_F)] \geq E[\min(k, X_I)]$ . This fact follows immediately, for example, from Corollary 4 of [León and Perron 2003]. Thus we have the following:

**COROLLARY 3.3.** *For any saturated instance of the adwords problem, the Hypothetical-Oblivious algorithm achieves a revenue of  $\sum_i B_i(1 - \sqrt{\frac{\gamma_i}{2\pi}})$ .*

---

**ALGORITHM 1:** Distribution independent algorithm for saturated instances

---

**Input** : Budgets  $B_i$  for  $i \in [n]$ , maximum possible bids  $b_i = \max_j b_{ij}$  for  $i \in [n]$  and the total number of queries  $m$

**Output:** An online assignment of queries to advertisers

- 1 Initialize  $R_i^0 = B_i$  for all  $i$ ;
  - 2 **for**  $t = 1$  to  $m$  **do**
    - (1) Let  $j$  be the query that arrives at time  $t$
    - (2) For each advertiser  $i$ , compute using Equation (3)
$$\Delta_i^t = \min(b_{ij}, R_i^{t-1}) + \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1} - \min(b_{ij}, R_i^{t-1}), m - t\right) - \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1}, m - t\right)$$
    - (3) Assign the query to the advertiser  $i^* = \operatorname{argmax}_{i \in [n]} \Delta_i^t$
    - (4) Set  $R_i^t = R_i^{t-1}$  for  $i \neq i^*$  and set  $R_{i^*}^t = R_{i^*}^{t-1} - \min(b_{i^* j}, R_{i^*}^{t-1})$
- 

### 3.2. A distribution independent algorithm for saturated instances

We now proceed to construct a distribution independent algorithm for saturated instances of the adwords problem that achieves at least as much revenue as achieved by the hypothetical algorithm Hypothetical-Oblivious. While our algorithm, given by Algorithm 1, remains the same for integral and fractional bids case, the argument is easier for integral bid case. Therefore, we begin with the integral case first.

**3.2.1. Single-bid instances (or Integral instances).** The idea of our algorithm is quite simple. When a query arrives, do the following: assuming that the Hypothetical-Oblivious algorithm will be implemented for the rest of steps, find the advertiser  $i$ , who when assigned the query will maximize the sum of the revenue obtained in this step together with the residual expected revenue that can be obtained in the remaining steps (where the residual expected revenue is calculated taking the remaining steps to be Hypothetical-Oblivious). Since the Hypothetical-Oblivious algorithm is just a balls and bins process that throws balls of value  $b_i$  into bin  $i$  with probability  $\frac{B_i}{b_i m}$ , the algorithm is the following: assuming that the remaining steps follow the simple and non-adaptive balls and bins process that throws a ball of value  $b_i$  into bin  $i$  with probability  $\frac{B_i}{b_i m}$ , compute the bin that when assigned the ball will maximize the sum of this step's revenue and expected residual revenue.

More formally, let  $j$  be the  $t$ -th query. We compute the difference in the expected revenue contributed by  $i$ , when it is assigned query  $j$  and when it is not assigned query  $j$ . That advertiser who maximizes this difference is assigned the query. The expected residual revenue  $\mathcal{R}(p, b, k, l)$  is a function of the following four quantities:

- the probability  $p$  with which a ball is thrown into this bin in the balls and bins process;
- the value  $b$  of each ball;
- the remaining space  $k$  in the bin;
- the remaining number of balls  $l$ ;

Let  $R_i^{t-1}$  denote the remaining budget of advertiser  $i$ , when the  $t$ -th query/ball arrives. Then for each advertiser  $i$ , we compute the difference

$$\Delta_i^t = \min(b_{ij}, R_i^{t-1}) + \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1} - \min(b_{ij}, R_i^{t-1}), m - t\right) - \mathcal{R}\left(\frac{B_i}{b_i m}, b_i, R_i^{t-1}, m - t\right),$$

and assign the query to the advertiser  $i^* \in \operatorname{argmax}_i \Delta_i^t$ . This is precisely what Algorithm 1 describes.

The quantity  $\mathcal{R}(p, b, k, l)$  is computed as follows. The residual expected revenue can be seen as the difference of two quantities

- the expected amount of balls to be aimed at the bin in the remaining steps =  $blp$ ;
- the expected amount of wasted balls, where waste occurs when  $\lceil \frac{k}{b} \rceil$  or more balls are thrown, and is given by  $\sum_{r=\lceil \frac{k}{b} \rceil}^l (rb - k) \binom{l}{r} p^r (1-p)^{l-r}$ .

Thus we have

$$\mathcal{R}(p, b, k, l) = blp - \sum_{r=\lceil \frac{k}{b} \rceil}^l (rb - k) \binom{l}{r} p^r (1-p)^{l-r} \quad (3)$$

LEMMA 3.4. *Algorithm 1 obtains an expected revenue at least as much as the Hypothetical-Oblivious algorithm.*

PROOF. We prove the lemma by a hybrid argument. Let  $A^r P^{m-r}$  represent a hybrid algorithm that runs our algorithm 1 for the first  $r$  steps, and the Hypothetical-Oblivious algorithm for the remaining  $m - r$  steps. If for all  $1 \leq r \leq m$  we prove that  $E[A^r P^{m-r}] \geq E[A^{r-1} P^{m-r+1}]$ , we would have proved that  $E[A^m] \geq E[P^m]$ , and thus the lemma. But  $E[A^r P^{m-r}] \geq E[A^{r-1} P^{m-r+1}]$  follows immediately from the definition of  $A$  which makes the expected revenue maximizing choice at any given step assuming Hypothetical-Oblivious on integer bids for the remaining steps.  $\square$

**3.2.2. Arbitrary bid instances (or fractional instances).** We now move on to fractional instances. Our algorithm is the same as for the integral instances, namely 1. However, the hybrid argument here is a bit more subtle. Let  $P_F^m$  denote running the Hypothetical-Oblivious algorithm on fractional bids for  $m$  steps, and let  $P_I^m$  denote the same for integral bids. That is,  $P_I^m$  is a balls and bins process where bin  $i$  receives a ball of value  $b_i$  with probability  $\frac{B_i}{b_i m}$  and zero otherwise, whereas  $P_F^m$  is a fractional balls and bins process where the expected amount of ball thrown into bin  $i$  is  $\frac{B_i}{m}$  as in the integral Hypothetical-Oblivious, but the value of the balls can be anywhere in  $[0, b_i]$ . Our proof is going to be that  $E[A^m] \geq E[P_I^m]$ . Note that we do not compare the quantities that one would want to compare on first thought: namely  $E[A^m]$  and  $E[P_F^m]$ , and the inequality could possibly go either way in this comparison. Instead we just compare our algorithm with Hypothetical-Oblivious working on integral instances. Since, we know from Corollary 3.2 we know that  $P_I^m$  has a good competitive ratio, it is enough to prove that  $E[A^m] \geq E[P_I^m]$ .

LEMMA 3.5.  $E[A^m] \geq E[P_I^m]$

PROOF. We prove this lemma, like Lemma 3.4, by a hybrid argument, albeit with two levels. Let  $A^r P_F P_I^{m-r-1}$  denote a hybrid which runs our algorithm 1 for the first  $r$  steps (on the actual instance, which might be fractional), followed by a single step of Hypothetical-Oblivious on the actual instance (again, this could be fractional), followed by  $m - r - 1$  steps of Hypothetical-Oblivious on the integral instance (an integral instance which throws the same expected amount of ball into a bin in a given step as the fractional instance does). For all  $1 \leq r \leq m$ , we prove that

$$E[A^r P_I^{m-r}] \geq E[A^{r-1} P_F P_I^{m-r}] \geq E[A^{r-1} P_I^{m-r+1}] \quad (4)$$

Chaining inequality (4) for all  $r$ , we get the Lemma.

The first inequality in (4) follows from the definition of  $A$  because it chooses that bin which maximizes the expected revenue assuming the remaining steps are integral Hypothetical-Oblivious.

The second inequality has a bin-by-bin proof. Fix a bin, and the proof follows from a discussion similar to that in Section 3.1.2, namely the maximum wastage occurs when the distribution of ball size is extreme, i.e.,  $b_i$  or zero. Formally let  $X_F^r$  be the random variable representing the amount thrown by fractional Hypothetical-Oblivious in the  $r$ -th step, and let the corresponding integral variable  $X_I^r$ . If  $\lambda_{r-1}$  is the remaining budget after  $r - 1$  steps of  $A$  in the given bin, we are interested in how  $E[\min(\lambda_{r-1}, X_F^r + \sum_{t=r+1}^m X_I^t)]$  compares with  $E[\min(\lambda_{r-1}, \sum_{t=r}^m X_I^t)]$ . Among all distributions with a fixed expectation,  $X_I^r$  is the distribution with extreme ball size and hence results in maximum wastage due to overflow (and hence minimum expected revenue), and thus we have the second inequality. This follows from Corollary 4 of [León and Perron 2003].

□

Lemma 3.5 together with Corollary 3.2 proves Theorem 2.7.

### 3.3. A tight example showing asymptotic optimality

We now show a simple example that shows that even when the distributions are known, no algorithm can give a  $1 - o(\sqrt{\gamma})$  approximation. Consider two advertisers 1 and 2. Advertiser 1 has a budget of  $2B$  and 2 has a budget of  $B$ . There are four types of queries: 0-query, 1-query, 2-query and 1-2 query.

- (1) The 0-query is worth nothing to both advertisers
- (2) The 1-query is worth 1 to advertiser 1 and zero to advertiser 2
- (3) The 2-query is worth 2 to advertiser 2 and zero to advertiser 1
- (4) The 12-query is worth 1 to advertiser 1 and 2 to advertiser 2

There are totally  $m$  queries that arrive online. The 2-query occurs with probability  $\frac{B}{2m}$ , the 1-query with probability  $\frac{B-\sqrt{B}}{m}$ , the 12-query with probability  $\frac{\sqrt{B}}{m}$  and the 0-query with remaining probability. Notice that the  $\gamma$  for this instance is  $\frac{1}{B}$ . Thus it is enough to show that a loss of  $\Theta(\sqrt{B})$  cannot be avoided.

First the distribution instance has  $\frac{B}{2}$  2-queries,  $B - \sqrt{B}$  1-queries,  $\sqrt{B}$  12-queries and remaining zero queries. This means that the distribution instance has a revenue of  $2B$ , which is our benchmark.

Now consider the 12 queries. By Chernoff bounds, with a constant probability at least a  $\Theta(\sqrt{B})$  of such queries occur in an instance. In such instances, at least a constant fraction of these 12-queries, i.e.,  $\Theta(\sqrt{B})$  12-queries, occur at such a point in the algorithm where,

- with a constant probability the remaining 2-queries could completely exhaust advertiser 2's budget,
- and with a constant probability the remaining 2-queries could fall short of the budget of advertiser 2 by  $\Theta(\sqrt{B})$

Note that by Chernoff bounds these events occur with a constant probability. This is the situation that confuses the algorithm. Whom to assign such a 12-query to? Giving it to advertiser 2 will fetch one unit of revenue more, but then with a constant probability situation 1 occurs in which case it is correct in hindsight to have assigned this query to advertiser 1, thus creating a loss of 1. On the other hand if the algorithm assigns this 12-query to advertiser 1, with a constant probability situation 2 occurs, thus making it necessary for each of these  $\Theta(\sqrt{B})$  queries to have been given to advertiser 2. Thus for  $\Theta(\sqrt{B})$  queries, there is a constant probability that the algorithm will lose one unit

of revenue irrespective of what it decides. This costs the algorithm a revenue loss of  $\Theta(\sqrt{B})$ , thus proving asymptotic tightness.

#### 4. GENERAL INSTANCES WITH CONSUMPTION INFORMATION

In this section, we describe our algorithm for arbitrary instances of adwords, assuming that the algorithm is given with the budget consumption from every advertiser, by some optimal algorithm for the distribution instance. Let  $C_i$  denote the budget consumption from advertiser  $i$  by the optimal solution. The  $C_i$ 's are part of the input to the algorithm. Note that  $C_i \leq B_i$  and  $\sum_i C_i \leq m$ .

##### 4.1. Hypothetical-Oblivious algorithm

Like in Section 3, before presenting our algorithm, we first describe the hypothetical Hypothetical-Oblivious algorithm and show that it also achieves a revenue of  $\sum_i C_i(1 - \sqrt{\frac{\gamma_i}{2\pi}}) = \sum_i C_i \left(1 - \sqrt{\frac{b_i}{2\pi B_i}}\right)$ .

**4.1.1. Hypothetical-Oblivious algorithm on single-bid instances.** Like in Section 3, we begin with the integral instances, i.e., advertiser  $i$  has bids of  $b_i$  or 0. Let  $k_i = \frac{B_i}{b_i}$ . To prove that Hypothetical-Oblivious achieves a revenue of  $\sum_i C_i(1 - \sqrt{\frac{\gamma_i}{2\pi}}) = \sum_i C_i \left(1 - \sqrt{\frac{b_i}{2\pi B_i}}\right)$ , it is enough to prove that the expected revenue from advertiser  $i$  is at least  $C_i(1 - \frac{k_i^{k_i}}{k_i!e^{k_i}}) \geq C_i(1 - \frac{1}{\sqrt{2\pi k_i}})$ .

Note that when interpreted as a balls and bins process, Hypothetical-Oblivious corresponds to throwing a ball of value  $b_i$  with probability  $\frac{C_i}{b_i m}$  into bin  $i$  at every step (and not with probability  $\frac{B_i}{b_i m}$ ). To prove that this process achieves a revenue of  $C_i(1 - \frac{k_i^{k_i}}{k_i!e^{k_i}})$  is equivalent to proving that the value of wasted balls is at most  $C_i \frac{k_i^{k_i}}{k_i!e^{k_i}}$ . Dropping subscripts, and setting  $b_i = 1$  since it doesn't affect analysis below, this means we have to prove that in a bin of capacity  $B$ , when a ball is thrown at every step with probability  $C/m$ , the expected number of wasted balls is at most  $\frac{B^B}{B!e^B} \times C$ , i.e.,  $\frac{B^{B+1}}{B!e^B} \times \frac{C}{B}$ . From Section 3 Lemma 3.1 we know that  $\frac{B^{B+1}}{B!e^B}$  is the expected number of wasted balls when a ball is thrown with probability  $B/m$  at every step. All we have to prove now is that when a ball is thrown with probability  $C/m$ , the expected number of wasted balls is at most  $C/B$  fraction of the same when the probability was  $B/m$ .

**LEMMA 4.1.** *In a balls and bins process where a given bin with capacity  $B$  receives a ball with probability  $C/m$  at each step, the expected number of balls in the given bin after  $m$  steps is at least  $(1 - \frac{B^B}{B!e^B}) \times C$ .*

**PROOF.** Let the random variable  $Y$  denote the number of balls wasted after  $m$  steps. Then our goal is to prove that  $E[Y] \leq \frac{B^B}{B!e^B} \times C = \frac{B^{B+1}}{B!e^B} \times \frac{C}{B}$ . We have

$$\begin{aligned} E[Y] &= \sum_{r=B+1}^m (r-B) \binom{m}{r} \left(\frac{C}{m}\right)^r \left(1 - \frac{C}{m}\right)^{m-r} \\ &= \sum_{r=B+1}^m (r-B) \binom{m}{r} \left(\frac{B}{m}\right)^r \left(1 - \frac{B}{m}\right)^{m-r} \times \left(\frac{C/m}{B/m}\right)^r \left(\frac{1-C/m}{1-B/m}\right)^{m-r} \end{aligned} \quad (5)$$

---

**ALGORITHM 2:** Partially distribution dependent algorithm for general instances

---

**Input** : Budgets  $B_i$  for  $i \in [n]$ , Consumption  $C_i$  for  $i \in [n]$ , maximum possible bids  $b_i = \max_j b_{ij}$  for  $i \in [n]$  and the total number of queries  $m$

**Output:** An online assignment of queries to advertisers

- 1 Initialize  $R_i^0 = B_i$  for all  $i$ ;
  - 2 **for**  $t = 1$  to  $m$  **do**
    - (1) Let  $j$  be the query that arrives at time  $t$
    - (2) For each advertiser  $i$ , compute using Equation (3)
$$\Delta_i^t = \min(b_{ij}, R_i^{t-1}) + \mathcal{R}\left(\frac{C_i}{b_i m}, b_i, R_i^{t-1} - \min(b_{ij}, R_i^{t-1}), m - t\right) - \mathcal{R}\left(\frac{C_i}{b_i m}, b_i, R_i^{t-1}, m - t\right)$$
    - (3) Assign the query to the advertiser  $i^* = \operatorname{argmax}_{i \in [n]} \Delta_i^t$
    - (4) Set  $R_i^t = R_i^{t-1}$  for  $i \neq i^*$  and set  $R_{i^*}^t = R_{i^*}^{t-1} - \min(b_{i^* j}, R_{i^*}^{t-1})$
- 

By Lemma 3.1 we know that

$$\sum_{r=B+1}^m (r-B) \binom{m}{r} \left(\frac{B}{m}\right)^r \left(1 - \frac{B}{m}\right)^{m-r} \leq \frac{B^{B+1}}{B!e^B}.$$

Thus, it's enough if we prove that for all  $r \geq B+1$ ,  $f(r) = \left(\frac{C/m}{B/m}\right)^r \left(\frac{1-C/m}{1-B/m}\right)^{m-r} \leq C/B$ . Since  $C \leq B$  and hence  $\left(\frac{C}{B}\right) \left(\frac{1-B/m}{1-C/m}\right) \leq 1$ , the function  $f(r)$  decreases with  $r$ , and thus it's enough to prove that  $f(B+1) \leq C/B$ .

$$\frac{f(B+1)}{C/B} = \left(\frac{C}{B}\right)^B \left(1 + \frac{B-C}{m-B}\right)^{m-(B+1)} \leq \left(\frac{C}{B}\right)^B \left(1 + \frac{B-C}{m-B}\right)^{m-B} \leq \left(\frac{C}{B}\right)^B e^{B-C}$$

We now have to prove that  $\left(\frac{C}{B}\right)^B e^{B-C} \leq 1$ . Let  $B = tC$ , and thus  $t \geq 1$ . Thus, what we need to prove is that  $e^{(t-1)C} \leq t^{tC}$  for  $t \geq 1$ . It is a straight forward exercise in calculus to prove that  $e^{t-1} \leq t^t$  for all  $t \geq 1$ .

Thus  $f(B+1) \leq C/B$ , and this proves the lemma.  $\square$

**4.1.2. Extending Hypothetical-Oblivious to general bids.** Extending Hypothetical-Oblivious to general bids is identical to the discussion in Section 3.1.2, and we omit it here.

## 4.2. An algorithm for general instances with $C_i$ 's as input

We now proceed to give an algorithm that has reduced dependence on distributions, namely, it uses just the  $C_i$ 's as inputs as against the entire distributional knowledge. The algorithm is very similar to the one presented in Section 3.2 and is presented below as Algorithm 2.

The only difference from Algorithm 1 is that the calculation of  $\Delta_i^t$  is done using a probability of  $\frac{C_i}{b_i m}$  instead of  $\frac{B_i}{b_i m}$ . Apart from this difference, the two algorithms are identical, and Lemmas analogous to Lemmas 3.4 and 3.5 can be proven, and combining them we get Theorem 2.2.

## 5. UNWEIGHTED INSTANCES WITHOUT HELP OF ANY DISTRIBUTIONAL PARAMETERS

In this section, we prove Theorem 2.8 for unweighted instances. The algorithm used here is the same as that used for saturated instances, namely Algorithm 1. Since the revenue of the distribution instance is  $\text{OPT} = \sum_i C_i$ , to prove Theorem 2.8, i.e., to show

an competitive ratio of  $1 - \sqrt{\frac{\pi}{2\pi} \frac{\sum_i B_i}{\text{OPT}}}$ , it is enough to show that the expected number of wasted queries in Algorithm 1 is at most  $\frac{\sum_i B_i}{\sqrt{2\pi k}}$  where  $k = 1/\gamma$ . But note that since the expected revenue of Hypothetical-Oblivious algorithm for a saturated instance is at least  $\sum_i B_i(1 - \frac{1}{\sqrt{2\pi k}})$ , the expected number of queries wasted by Hypothetical-Oblivious when working on a saturated unweighted instance is at most  $\frac{\sum_i B_i}{\sqrt{2\pi k}}$ . We now show that this quantity is an upper bound on the expected number of queries wasted by our algorithm (Algorithm 1) for any unweighted instance (saturated or unsaturated).

**LEMMA 5.1.** *For any unweighted instance, the expected number of queries wasted by Algorithm 1 is at most the expected number of queries wasted by Hypothetical-Oblivious for a saturated instance.*

**PROOF.** We prove this lemma by a hybrid argument. Let  $P_B$  denote the Hypothetical-Oblivious algorithm for a saturated instance, and let  $P_C$  denote the Hypothetical-Oblivious algorithm for our instance (which might be unsaturated). Let  $A^r P_B^{m-r}$  represent a hybrid algorithm that runs our algorithm 1 on the given instance for the first  $r$  steps, and the Hypothetical-Oblivious algorithm on the saturated instance for the remaining  $m - r$  steps. Note that  $A$  and  $P_B$  are not just two different algorithms, but the instances they work on are also different. Let  $A^{r-1} P_C P_B^{m-r}$  represent a hybrid algorithm that runs our algorithm 1 on the given instance for the first  $r-1$  steps, and the Hypothetical-Oblivious algorithm on the given instance for the next step, and Hypothetical-Oblivious algorithm on a saturated instance for the remaining  $m - r$  steps. Let  $W[A^r P_B^{m-r}]$  denote the expected number of wasted balls when we run  $A^r P_B^{m-r}$ . For all  $1 \leq r \leq m$ , we prove that

$$W[A^r P_B^{m-r}] \leq W[A^{r-1} P_C P_B^{m-r}] \leq W[A^{r-1} P_B^{m-r+1}].$$

Chaining these inequalities for all  $r$ , we get the lemma. The first inequality follows by the definition of  $A$  because, it maximizes expected revenue assuming the rest of the steps follow Hypothetical-Oblivious on a saturated instance. This also means that  $A$  minimizes the expected number of wasted queries assuming the rest of the steps follow Hypothetical-Oblivious on a saturated instance. The second inequality follows trivially because the expected number of wasted queries is no smaller for Hypothetical-Oblivious on a saturated instance, than on an unsaturated instance.  $\square$

Lemma 5.1 proves Theorem 2.8.

## 6. ADVERSARIAL STOCHASTIC INPUTS

One feature of our algorithm is that the underlying distribution from which queries are drawn need not be the same throughout the algorithm. Even if the distribution changes over time, all our competitive ratios continue to hold, as long as the distributions are not “bad”. This notion of time varying distributions was introduced in [Devanur et al. 2011] and was called as the Adversarial stochastic input model (ASI). In the ASI model, at every step an adversary picks the distribution from which a query is to be drawn. The adversary can tailor the distribution based on how the algorithm has performed so far, but it is bound to pick only distributions whose distribution instance have a minimum consumption  $C_i$  from advertiser  $i$ . This consumption guarantee is the benchmark  $\sum_i C_i$  with respect to which the competitive ratio is defined. The algorithm, as before, is given the consumption information  $C_i$ , and the maximum bid  $b_i$  possible for each advertiser.

We show here for one of our theorems, how the competitive ratio holds under ASI model. Other results can be similarly extended and we skip the proofs here. Consider

the saturated instances setting in Section 3. The ASI model for this setting simply means that the adversary is free to pick any distribution at every step, but the distribution instance must have an optimal solution that saturates budgets. Our algorithm 1 picks the advertiser in each step who maximizes the expected revenue assuming that the remaining steps look like a balls and bins process where bin  $i$  receives a ball with probability  $B_i/m$ . Note that the assumption our algorithm is making is just related to the budget consumption  $B_i$  and is therefore oblivious to the exact distribution that can guarantee a budget consumption of  $B_i$ . A similar argument can be extended for the general consumption  $C_i$  case: the algorithm is oblivious to the exact distribution that realizes a consumption of  $C_i$  for the distribution instance.

## 7. CONCLUSION

In this paper, we consider the adwords problem in the unknown distribution model and provide improved competitive ratios for the entire range of budget-to-bid ratios. A significant aspect of our results is that asymptotically, the guarantee we get matches the best guarantee possible even with known distributions, while requiring significantly less information about the distribution, as we only need to know the  $C_i$ 's. With this, we almost close the last few remaining gaps for the adwords problem in the unknown distribution model. Some small gaps still remain, however. The most interesting of these is to remove the assumption that the algorithm needs to know the  $C_i$ 's. In fact the algorithm for the saturated instances should just work even for unsaturated instances. The intuition behind this is that the saturated instances are in fact the worst cases for this problem. The danger that the algorithm has to avoid is that of “overflow” and the resulting wastage, that is, assigning too much to a given advertiser so that some of the queries have to be then wasted. An unsaturated instance has less “stuff” to assign to begin with, in comparison to a saturated instance. Thus the danger of overflow should only be smaller for an unsaturated instance. The inability to extend our analysis to unsaturated instances seems to be more due to technical reasons than due to something fundamental. Hence we make the following conjecture.

**CONJECTURE 7.1.** *The adwords problem has an online algorithm that achieves a competitive ratio of  $1 - O(\sqrt{\gamma})$  for the stochastic unknown distribution model.*

A weaker conjecture is that Theorem 2.8 can be extended to the weighted case. The proof of Lemma 5.1 relied on counting the revenue through the expected number of wasted queries. This worked because each query was worth either 1 or zero (or more generally  $b$  or zero). Such an accounting scheme will not work for general instances where different advertisers could value queries at different non-zero values. The question is can we still get the ratio obtained for Theorem 2.8. We conjecture that this is possible.

**CONJECTURE 7.2.** *The adwords problem has an online algorithm that achieves a competitive ratio of  $1 - \sqrt{\frac{\gamma}{2\pi} \frac{\sum_i B_i}{\text{OPT}}}$  for the stochastic unknown distribution model.*

## REFERENCES

- BAHMANI, B. AND KAPRALOV, M. 2010. Improved bounds for online stochastic matching. In *ESA*. 170–181.
- BUCHBINDER, N., JAIN, K., AND NAOR, J. S. 2007. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA'07: Proceedings of the 15th annual European conference on Algorithms*. Springer-Verlag, Berlin, Heidelberg, 253–264.
- CHARLES, D., CHICKERING, M., DEVANUR, N. R., JAIN, K., AND SANGHI, M. 2010. Fast algorithms for finding matchings in lopsided bipartite graphs with applications



- to display ads. In *EC '10: Proceedings of the 11th ACM conference on Electronic commerce*. ACM, New York, NY, USA, 121–128.
- DEVANUR, N. R. AND HAYES, T. P. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *ACM Conference on Electronic Commerce*, J. Chuang, L. Fortnow, and P. Pu, Eds. ACM, 71–78.
- DEVANUR, N. R., JAIN, K., SIVAN, B., AND WILKENS, C. A. 2011. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*. EC '11. ACM, New York, NY, USA, 29–38.
- FELDMAN, J., MEHTA, A., MIRROKNI, V., AND MUTHUKRISHNAN, S. 2009. Online stochastic matching: Beating  $1-1/e$ . In *FOCS '09: Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, Washington, DC, USA, 117–126.
- HAEUPLER, B., MIRROKNI, V. S., AND ZADIMOUGHADDAM, M. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*. 170–181.
- KALYANASUNDARAM, B. AND PRUHS, K. 1998. On-line network optimization problems. In *Developments from a June 1996 seminar on Online algorithms: the state of the art*. Springer-Verlag, London, UK, 268–280.
- KARANDE, C., MEHTA, A., AND TRIPATHI, P. 2011. Online bipartite matching with unknown distributions. In *STOC*. 587–596.
- LEÓN, C. A. AND PERRON, F. 2003. Extremal properties of sums of bernoulli random variables. *Statistics & Probability Letters* 62, 4, 345–354.
- MAHDIAN, M. AND YAN, Q. 2011. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *STOC*.
- MANSHADI, V. H., GHARAN, S. O., AND SABERI, A. 2011. Online stochastic matching: Online actions based on offline statistics. In *SODA*. 1285–1294.
- MEHTA, A., SABERI, A., VAZIRANI, U., AND VAZIRANI, V. 2005. Adwords and generalized on-line matching. In *FOCS 05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 264–273.
- MIRROKNI, V., GHARAN, S. O., AND ZADIMOUGHADDAM, M. 2012. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *SODA*.
- YAN, Q. 2011. Mechanism design via correlation gap. In *SODA*. 710–719.