

# Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems

Yunhong Zhou  
HP Labs  
1501 Page Mill Rd  
Palo Alto CA 95304  
yunhong.zhou@hp.com

Deeparnab Chakrabarty\*  
College of Computing  
Georgia Tech  
Atlanta, GA 30332  
deepc@cc.gatech.edu

Rajan Lukose  
HP Labs  
1501 Page Mill Rd  
Palo Alto CA 95304  
rajan.lukose@hp.com

## ABSTRACT

We consider the budget-constrained bidding optimization problem for sponsored search auctions, and model it as an *online (multiple-choice) knapsack problem*. We design both deterministic and randomized algorithms for the online (multiple-choice) knapsack problems achieving a *provably optimal* competitive ratio. This translates back to fully automatic bidding strategies maximizing either profit or revenue for the budget-constrained advertiser. Our bidding strategy for revenue maximization is oblivious (i.e., without knowledge) of other bidders' prices and/or click-through-rates for those positions. We evaluate our bidding algorithms using both synthetic data and real bidding data gathered manually, and also discuss a sniping heuristic that strictly improves bidding performance. With sniping and parameter tuning enabled, our bidding algorithms can achieve a performance ratio above 90% against the optimum by the omniscient bidder.

**Categories and Subject Descriptors:** G.1.6 [Mathematics of Computing]: Mathematical Software - Algorithm Design and Analysis

**General Terms:** Algorithms, Theory, Economics

**Keywords:** Sponsored search auction, keyword bidding, online knapsack problem, multiple-choice knapsack problem

## 1. INTRODUCTION

Sponsored search auction is an effective way of monetizing search query activities for the search engine provider, while shifting the burden to the advertiser to figure out how to automate and optimize the keyword bidding process. In this work we focus on the bid optimization question under the budget constraint.

**Keyword Bidding Models.** For simplicity, assume that the default advertiser has a budget  $B$  over a fixed time horizon, discretized into time periods  $1, \dots, T$ . He is interested in a single keyword with expected value-per-click  $V$ .

There are bidders  $\{1, \dots, N\}$  at time  $t$  for this keyword and their bids are sorted in decreasing order  $b_1(t) > \dots > b_N(t)$ . There are  $S$  ad slots, and are assigned to the top- $S$  bids as follows: bidder  $s$  gets slot  $s$ ; for each user click on his ad, bidder  $s$  is charged a price  $b_{s+1}$ , if  $s < S$  or a minimum fee  $b_{\min}$  (usually 10¢). Each slot  $s$  has a click-through-rate

\*Work was done while the author was an intern at HP Labs.

(CTR), denoted  $\alpha(s)$ , which is defined as the total number of clicks on an ad divided by the total number of impressions (displays). Therefore the default advertiser can obtain slot  $s$  by bidding slightly over  $b_s(t)$ ; for each user click, he incurs a cost of  $b_s(t)$ , obtains an expected revenue  $V$  and profit  $V - b_s(t)$ .

**Online Knapsack Problems.** Let us start with the relatively simple single-slot case. At time  $t$ ,  $b(t)$  is the maximum bid among bidders 1 to  $N$ , and  $X(t)$  is the number of clicks at period  $t$ . Winning at time  $t$  costs the advertiser  $w(t)$  and earns him a profit of  $v(t)$  where

$$w(t) \equiv b(t)X(t)\alpha, \quad v(t) \equiv (V - b(t))X(t)\alpha. \quad (1)$$

For revenue maximization,  $v(t) = VX(t)\alpha$ . Since the default bidder has to decide either overbidding  $b(t)$  or not at time  $t$ , thus keyword bidding corresponds to the *online* knapsack problem (Online-KP). The case of multiple slots is captured by the online version of the multiple-choice knapsack problem (Online-MCKP), where each time the advertiser can win at most one ad slot.

**Our Assumptions.** In general, no online algorithm can achieve any non-trivial *competitive ratio* (the ratio between the output of the given algorithm and the offline optimum) for Online-KP [4]. Fortunately, in our setting, we make two reasonable assumptions on the knapsack items, which allow us to develop interesting online algorithms. These two assumptions are:

$$(i) w(t) \ll B; \quad (ii) L \leq \frac{v(t)}{w(t)} \leq U, \quad \forall t. \quad (2)$$

## 2. RELATED WORK

**Keyword Bidding.** Sponsored search auctions have attracted a lot of attention, for both auctioneer revenue maximization and advertiser bidding optimization. Among all these work, Mehta et al. [5] studied the auctioneer revenue maximization with budget-constrained bidders, using a trade-off function  $\Psi$  (compare it to our threshold function) to grant queries to bidders, and the technique they use is probably most similar to the threshold function we use.

**Online Algorithms.** Awerbuch et al. [2] studied the online call routing which generalizes the online classical knapsack problem. More recently, Buchbinder et al. [3] designed online algorithms for fractional versions of general packing problems which imply an  $O(\ln(U/L))$ -competitive algorithm for the online knapsack problem.

### 3. THE ONLINE KNAPSACK PROBLEM

In this section we design competitive algorithms for the Online-KP with assumptions (2)(i),(ii). We describe two  $(\ln(U/L) + 1)$ -competitive algorithms for the problem.

**Randomized Algorithm:** Let  $\mathcal{D}$  be the continuous distribution from 0 to  $U$ , with the following density function:  $f(x) = \frac{c}{x}$ , for  $L \leq x \leq U$ , and  $f(x) = c/L$  for  $0 \leq x \leq L$ , where  $c = \frac{1}{1 + \ln(U/L)}$ . Algorithm ONLINE-KP-RANDOMIZED simply picks a threshold  $\rho$  from the distribution  $\mathcal{D}$ ; at each time, it picks item  $t$  iff its efficiency is at least  $\rho$  and its budget allows.

**Deterministic Algorithm:** Our deterministic algorithm for Online-KP works against all adversaries..

#### Algorithm ONLINE-KP-THRESHOLD

Let  $\Psi(z) \equiv (Ue/L)^z(L/e)$ .

At time  $t$ , let  $z(t)$  be the fraction of capacity filled, pick element  $t$  iff  $\frac{v(t)}{w(t)} \geq \Psi(z(t))$ .

**THEOREM 3.1.** Both ONLINE-KP-RANDOMIZED and ONLINE-KP-THRESHOLD have a competitive ratio of  $\ln(U/L) + 1$ .

A matching lower bound:

**THEOREM 3.2.** The competitive ratio of any (possibly randomized) online algorithm for the online knapsack problem is at least  $\ln(U/L) + 1$ .

### 3.1 Single-Slot Auctions

we can translate the algorithms ONLINE-KP-THRESHOLD to bidding strategies for single-slot keyword auctions for both profit and revenue maximization, based on Eq. (1). Details are omitted due to space constraints.

### 4. ONLINE-MCKP & MULTIPLE SLOTS

**The Online-MCKP.** The Online-MCKP is a generalization of the Online-KP. At each time period, at most one item can be selected from the item-set  $N_t = \{(v_s(t), w_s(t))\}$ . The goal again is to maximize the total value of items selected.

The Algorithm Online-MCKP-Threshold is a generalization of Alg Online-KP-Threshold, and it works as follows: Let  $\Psi(z)$  and  $z(t)$  be the same as before. At time  $t$ , let  $E_t \equiv \left\{ s \in N_t \mid \frac{v_s(t)}{w_s(t)} \geq \Psi(z(t)) \right\}$ . If  $E_t \neq \emptyset$ , pick element  $s \in E_t$  with maximum  $v_s(t)$ .

**THEOREM 4.1.** ONLINE-MCKP-THRESHOLD has a competitive ratio of  $(\ln(U/L) + 2)$ .

**Multiple-Slot Bidding.** Next we translate Alg. ONLINE-MCKP-THRESHOLD to bidding strategies for both profit and revenue maximization. It turns out the revenue maximization strategy is the same as the single-slot case, thus omitted. The profit-maximization bidding strategy is below:

#### Bidding Strategy

PROFIT-MAXIMIZING MULTIPLE-SLOT

Fix  $\epsilon > 0$ . Let  $\Psi(z) \equiv (Ue/\epsilon)^z(\epsilon/e)$ .

At time  $t$ , let  $z(t)$  be fraction of budget spent,

$$E_t \equiv \left\{ s \mid b_s(t) \leq \frac{V}{1 + \Psi(z(t))} \right\},$$

bid  $b_s(t)$  where  $s = \arg \max_{s \in E_t} (V - b_s(t))\alpha(s)$ .

### 5. EXPERIMENTAL EXPLORATION

In this section, we evaluate our bidding algorithms using both synthetic and real-world data, and discuss two useful heuristics: sniping and parameter tuning.

**The Sniping Heuristic.** Simulation of the bidding algorithm described above with synthetic dataset shows that the bidding algorithm is too conservative, and leaves a significant fraction of budget unspent. Thus a potential performance improvement is *sniping* towards the end of the auction. If the bidder has knowledge (reliable estimates) about the click traffic ( $X(t)$ ) and the click-through rates, then the bidding strategies can be modified to increase its bid appropriately. Theoretically, we can prove that *sniping* strictly improve the performance of the algorithm. For details, see the technical report [6].

**Parameter Tuning.** If the lower bound  $L$  in the online knapsack problem is too small, we can replace it with a larger value  $L' > L$  for the threshold function  $\Psi$ . This essentially discards items with very low efficiency, and the loss is minimal if the optimal solution consists of items with relatively high efficiency. It turns out tuning the parameter  $L$  makes a significant performance improvement.

**Evaluation using Real Bidding Data.** Next we report some experimental results on evaluating bidding algorithms for multiple-slot auctions using real bidding data. We scraped bidding data from the now defunct Overture webpage [1] with continuous crawling for about two weeks, for one of the most dynamic and expensive keyword "auto insurance." There are totally  $T = 1842$  distinct time periods in our collected data, and most top-5 bids are larger than \$10. For the experiments, we use  $B = 1000$ , and three different values  $V = 8, 10, 12$ . We evaluated both the profit-maximizing and revenue-maximizing strategies with and without sniping. For all these experiments, we use  $U = V/b_{\min} - 1$  for profit maximization and  $U = V/b_{\min}$  for revenue maximization, and  $b_{\min} = 0.9$ .

Since results are very similar for different parameter values, we summarize them in Table 5. For all the examples we run, sniping improves the bidding performance significantly while exhausting the budget. Table 5 seems to tell us, for almost all values, with parameter tuning of  $L$ , the performance ratio (ALG/OPT) is around 70%-75% without sniping, and 90%-95% with sniping.

Profit-Maximization Bidding Performance						
V	OPT	ALG	ALG/ OPT	budget left	ALG (sniping)	ALG/ OPT
8	3779	2751	73%	225.5	3541	94%
10	4974	4059	82%	116.1	4607	93%
12	6169	4463	72%	240.8	5842	95%

### 6. REFERENCES

- [1] Overture view bids, available until February 2007. <http://uv.bidtool.overture.com/d/search/tools/bidtool>
- [2] B. Awerbuch, Y. Azar, and S. A. Plotkin. Throughput-competitive on-line routing. In *FOCS*, pages 32–40, 1993.
- [3] N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proc. ESA*, pages 253–264, 2007.
- [4] A. Marchetti-Spaccamela and C. Vercellis. Stochastic on-line knapsack problems. *Math. Programming*, 68:73–104, 1995.
- [5] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. Adwords and generalized on-line matching. In *Proc. FOCS*, pages 264–273, 2005.
- [6] Y. Zhou, D. Chakrabarty, and R. Lukose. Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems. HP Labs tech report HPL-2008-9, 2008.