Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems*

Yunhong Zhou[1], Deeparnab Chakrabarty[2], and Rajan Lukose[3]

[1] Rocket Fuel Inc, Redwood Shores, CA
yzhou@rocketfuelinc.com
[2] Georgia Tech, Atlanta, GA
deepc@cc.gatech.edu
[3] HP Labs, Palo Alto, CA
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 clickthrough-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.

1 Introduction

Sponsored search auction is an effective way of monetizing search query activites for search engine providers, while shifting the burden to advertisers/bidders to figure out how to automate and optimize the keyword bidding process. In this work we focus on the bid optimization problem under the budget constraint. Formally, given an advertiser with a fixed budget over a fixed time horizon, and a set of keywords that he is interested to bid on, we try to design bidding strategies to address the following problem: For each keyword and each time period, how much should the advertiser bid to obtain which position, so as to maximize return on investment (ROI) of these auctions?

Keyword Bidding Models. For simplicity, assume that the default advertiser has a budget B over a fixed time horizon, discretized into time periods $1, \ldots, T$. He is interested in a single keyword with expected value-per-click V. The model can be easily extend to the multiple-keyword case.

* Work was done while the author were at HP Labs.

C. Papadimitriou and S. Zhang (Eds.): WINE 2008, LNCS 5385, pp. 566–576, 2008.

There are bidders $\{1, \cdots, N\}$ at time t for this keyword and their bids are sorted in decreasing order $b_1(t) > \ldots > b_N(t)$. 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 $b_{\min}$ (e.g. 10¢). Each slot s has a clickthrough-rate (CTR), denoted $\alpha(s)$, which is defined as the total number of clicks on an ad divided by the total number of impressions (displays). Assuming other bidders have their bids fixed, 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. Fix a keyword with positions $1, \ldots, S$. At time t, $X(t)$ is the number of clicks at period t, while $b_s(t)$ is the maximum bid corresponding to position s. Winning position s at time t costs the advertiser $w_s(t)$ and earns him a profit of $v_s(t)$ where

$$w_s(t) \equiv b_s(t)X(t)\alpha(s), \qquad v_s(t) \equiv (V - b_s(t))X(t)\alpha(s). \tag{1}$$

For revenue maximization, $v_s(t) = VX(t)\alpha(s)$. Let $N(t) = \{(w_s(t), v_s(t)) | s = 1, \ldots, S\}$, then winning position s at time t correspondes to selecting item $(w_s(t), v_s(t)) \in N_t$. Since the default bidder has to decide either overbidding $b_s(t)$ or not at time t, thus keyword bidding corresponds to the *online* multipe-choice knapsack problem (Online-MCKP). The multiple-choice knapsack problem is a generalization of the classic knapsack problem, where there are multiple item-sets and you can select *at most one* item from each item-set; the multiple-choice constraint of MCKP corresponds to the sponsored search auction policy where each advertiser can select to win at most one ad slot for each keyword at each time.

Our Assumptions. We use *competitive analysis* to evaluate our bidding strategies, comparing our result with the maximum profit attainable by the *omniscient bidder* who knows the bids of all the other users ahead of time. 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:

$$\text{(i)} w_s(t) \ll B; \qquad \text{(ii)} L \leq \frac{v_s(t)}{w_s(t)} \leq U, \quad \forall t,\ \forall s. \tag{2}$$

2 Results

In this work we model budget-constrained bidding optimization as variants of online knapsack problems. In Section 3, we design a determinstic algorithm for the online knapsack problem with two assumptions given above. The algorithm has a competitive ratio $\ln(U/L) + 1$, and is robust again any adaptive adversary. We also show a *matching lower bound* in section 3.1. Therefore our algorithm is *provably optimal* in the worst-case sense. We also give a $(\ln(U/L)+2)$-competitive online algorithm for the multiple-choice knapsack problem (MCKP), the classic generalization of the knapsack problem which corresponds to the general bidding optimization problem with multiple slots per keyword.

In Section 4, we translate the algorithms for online knapsack problems into bidding strategies for sponsored search auctions, for both profit and revenue maximization. For single-slot auctions, the corresponding strategies are *oblivious*, and thus work even if other bidders' bids were not known. It also implies that the strategy is an *approximate dominant strategy* in the sense that it is an approximate best response to *any* bid profile of other bidders. For the multiple-slot case, we translate the algorithm for Online-MCKP to bidding strategies for both profit and revenue-maximizing bidding strategies. The profit maximizing strategy is not oblivious and requires knowledge of other players' bids and also the CTRs of all slots. The revenue-maximizing strategy remains oblivious.

In Section 5, we report experimental results evaluating our bidding strategies using both synthetic bidding data and real bidding data collected manually. We modify our strategy by adding a *sniping* heuristic, and it performs much better empirically while maintaining the same theoretical bounds. Our limited experimental evaluation also suggests that parameter tuning helps to improve the performance of our bidding algorithms. With both sniping and parameter tuning enabled, our bidding algorithms (for both profit and revenue maximization) achieve an output value which is consistently more than 90% of the optimum by the omniscient bidder.

2.1 Related Work

Due to page limit as well as the vast amount of research literature in sponsored search auctions, knapsack problems, and online algorithms, we will only discuss previous work most relavant to ours.

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 etc al. [5] studied the auctioneer revenue maximization with budget-constrained bidders, using a trade-off function Ψ (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 Online Knapsack Problems

Consider the online version of the classic 0/1 knapsack problem. The input sequence consists of a knapsack of capacity B and a stream of T items where item t has value $v(t)$ and weight $w(t)$. We call the value-to-weight ratio $v(t)/w(t)$ of item t its *efficiency*. The goal is to choose these items in an *online* fashion, i.e., making a decision as an item arrives and not revoking them later, so as to maximize the total value of selected items. For the online *multiple-choice* knapsack problem, at each step a set of items N_t arrives and we need to choose at most one item from each set.

We say that an online algorithm $\mathcal{A}$ has competitive ratio γ (or equivalently is γ-competitive) if for *any* input sequence σ, we have $\mathrm{OPT}(\sigma) \leq \gamma \cdot \mathcal{A}(\sigma)$, where $\mathcal{A}(\sigma)$ is the (expected, if $\mathcal{A}$ is randomized) value obtained by $\mathcal{A}$ given σ, and $\mathrm{OPT}(\sigma)$ is the maximum value which can be obtained by any *offline* algorithm with the knowledge of σ.

We now give a deterministic algorithm for the online knapsack problem achieving the optimal bound of $\ln(U/L) + 1$. In the remainder of the paper, e denotes the base of the natural logarithm.

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

$$\frac{v(t)}{w(t)} \geq \Psi(z(t)).$$

Observe that for $z \in [0, c]$ where $c \equiv 1/(1 + \ln(U/L))$, $\Psi(z) \leq L$, thus the algorithm will pick all items available until c fraction of the knapsack is filled. In fact, we will assume henceforth $\Psi(z) = L$ for $z \in [0, c]$. When $z = 1$, $\Psi(z) = U$, and since Ψ is strictly increasing, the algorithm will never over-fill the knapsack.

Theorem 1. Online-KP-Threshold *has a competitive ratio of* $\ln(U/L) + 1$.

Proof. Fix an input sequence σ. Let the algorithm terminate filling Z fraction of the knapsack and obtaining a value of $\mathcal{A}(\sigma)$. Let S and S^* respectively be the set of items picked by the Algorithm Online-KP-Threshold and the optimum. Denote the weight and the value of the common items by $W = w(S \cap S^*)$ and $P = v(S \cap S^*)$. For each item t not picked by the algorithm, its efficiency is $< \Psi(z(t)) \leq \Psi(Z)$ since $\Psi(z)$ is a monotone increasing function of z. Thus,

$$\mathrm{OPT}(\sigma) \leq \mathrm{P} + \Psi(\mathrm{Z})(\mathrm{B} - \mathrm{W})$$

Since $\mathcal{A}(\sigma) = P + v(S \setminus S^*)$, the above inequality implies that

$$\frac{\mathrm{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{P + \Psi(Z)(B - W)}{P + v(S \setminus S^*)}. \tag{3}$$

Since each item j picked in S must have efficiency at least $\Psi(z_j)$ where z_j is the fraction of the knapsack filled at that instant, we have

$$P \geq \sum_{j \in S \cap S^*} \Psi(z_j) w_j =: P_1, \tag{4}$$

$$v(S \setminus S^*) \geq \sum_{j \in S \setminus S^*} \Psi(z_j) w_j =: P_2. \tag{5}$$

Since $\mathrm{OPT}(\sigma) \geq \mathcal{A}(\sigma)$, inequality (3) implies

$$\frac{\mathrm{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{P + \Psi(Z)(B - W)}{P + v(S \setminus S^*)} \leq \frac{P_1 + \Psi(Z)(B - W)}{P_1 + v(S \setminus S^*)} \tag{6}$$

Since $P_1 \leq \Psi(Z) w(S \cap S^*) = \Psi(Z) W$, plugging in the values of P_1 and P_2, we get

$$\frac{\mathrm{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{\Psi(Z)}{\sum_{j \in S} \Psi(z_j) \Delta z_j} \tag{7}$$

where $\Delta z_j = z_{j+1} - z_j = w_j / B$ for all j.

Based on the assumption that the weights are much smaller than B, we can approximate the summation via an integration (refer to the remark following the proof). Thus,

$$\begin{aligned} \sum_{j \in S} \Psi(z_j) \Delta z_j \approx \int_0^Z \Psi(z) dz &= \int_0^c L dz + \int_c^Z \Psi(z) dz \\ &= cL + \frac{L}{e} \frac{(Ue/L)^Z - (Ue/L)^c}{\ln(Ue/L)} \\ &= \frac{L}{e} \frac{(Ue/L)^Z}{\ln(Ue/L)} = \frac{\Psi(Z)}{\ln(U/L) + 1} . \end{aligned}$$

Along with inequality (7), this completes the proof.

Remark: We can make the approximation made above precise. Since $\Psi(z)$ is an increasing function of z, we obtain $\sum_{j \in S} \Psi(z_j) \Delta z_j \geq (1 - \epsilon_0) \int_0^Z \Psi(z) dz$ where $\epsilon_0 = (\max_j w_j)/B$ is small constant. Thus, to be precise, the competitive ratio is actually $\ln(Ue/L) \cdot \frac{1}{1 - \epsilon_0}$. For simplicity, we ignore the factor $1 - \epsilon_0$ for subsequent analysis.

Extension to Online-MCKP. One can extend the above algorithm to multiple choice knapsack problems in the following way – at each step t, let $E_t \subseteq N_t$ denote the items with efficiency at least $\Psi(z(t))$. Pick the item in E_t with the highest profit. Call this algorithm Online-MCKP-Threshold. The following theorem can be proved similarly as above and we omit it from the extended abstract.

Theorem 2. Online-MCKP-Threshold *has a competitive ratio of* (ln $(U/L) + 2)$.

3.1 A Matching Lower Bound

Theorem 3. *The competitive ratio of any (possibly randomized) online algorithm for the online knapsack problem is at least* $\ln(U/L) + 1$.

Proof (Sketch). The proof constructs a distribution of input sequences and shows that any deterministic algorithm against this distribution achieves a competitive ratio at most $\ln(U/L)+1$. Based on Yao's minimax lemma [6], the lower bound is obtained. We describe the distribution here and defer the analysis to our technical report.

Fix a parameter $\eta > 0$. Let k be an integer such that $(1+\eta)^k = U/L$, i.e., $k = \frac{\ln(U/L)}{\ln(1+\eta)}$. The support of the input distribution consists of the instances $I_0, I_1, \cdots, I_k$, where I_0 is a stream of B identical items each with weight 1 and value L. I_1 is I_0 followed by a stream of B identical items each with weight 1 and value $(1+\eta)L$, and in general I_{j+1} is I_j followed by B items with weight 1 and value $(1+\eta)^{j+1}L$. The distribution $\mathcal{D}$ is specified by giving probability p_j to instance I_j where

$$p_k := \frac{1+\eta}{(k+1)\eta+1}, \quad p_j := \frac{\eta}{(k+1)\eta+1}, \ \forall\ 0 \le j < k.$$

The ratio is obtained as $\eta \to 0$.

4 Bidding Strategies for Keyword Auctions

In this section, we construct bidding strategies for either profit maximization or revenue maximization. The difference in the two are in the parameter settings. For simplicity and brevity, we start with the single-slot case and extend to the multiple-slot case.

For profit maximization, recall that outbidding $b(t)$ at time t gives an *efficiency* of $\frac{v(t)}{w(t)} = \frac{V}{b(t)} - 1$ while for revenue maximization its $\frac{V}{b(t)}$. Thus, the parameters U and L for revenue maximization strategies are: $U_r := \frac{V}{b_{min}}$ and $L_r := 1$ respectively. For profit maximization $U_p = U_r - 1$, though L_p could be 0. To take care of this, we introduce another parameter ϵ, such that we bid only when the efficiency is bigger than ϵ. This makes $L_p = \epsilon$ but leads to an *additive loss* in the performance guarantee.

The strategies are derived from the online algorithms: Bidder 0 outbids only if the efficiency is bigger than the threshold. Since the threshold does not depend on anything other than the fraction of knapsack filled, the strategies also depend only on the fraction of budget spent. The strategies are formally stated as follows:

Bidding Strategy: PROFIT-MAXIMIZING SINGLE-SLOT
Let $\Psi(z) \equiv (U_p e/\epsilon)^z (\epsilon/e)$.
At time t, if fraction of budget spent is $z(t)$, then bid $b_0(t) = \frac{V}{1+\Psi(z(t))}$.

Bidding Strategy: REVENUE-MAXIMIZING SINGLE-SLOT
Let $\Psi(z) \equiv (U_r e)^z (1/e)$.
At time t, if fraction of budget spent is $z(t)$, then bid $b_0(t) = \frac{V}{1+\Psi(z(t))}$

Notice that both strategies only need the fraction of budget spent and are thus oblivious to the other parameters of the auction. We use Profit and Revenue to denote the profit and revenue earned by the above strategies respectively, and $\mathrm{OPT_p}$ and $\mathrm{OPT_r}$ to denote the profit and revenue of an omniscient bidder. Then we have the following theorem:

Theorem 4. *(i) For single-slot profit maximization, for any $\epsilon > 0$,*

$$\mathrm{OPT_p} \leq \epsilon \mathrm{B} + \left(\ln\left(\frac{(\mathrm{V} - \mathrm{b_{min}})}{\epsilon \mathrm{b_{min}}}\right) + 1\right) \text{ Profit.}$$

(ii) For single-slot revenue maximization, assuming that OPT *does not overbid at time t where $b(t) > V$,*

$$\mathrm{OPT} \leq \left(\ln\left(\frac{\mathrm{V}}{\mathrm{b_{min}}}\right) + 1\right) \text{ Revenue.}$$

The proof of Theorem 4(i) follows from Theorem 1 and the fact that all items with efficiency $\leq \epsilon$ has total value at most ϵB. Theorem 4 also suggests that different ϵ values give different guarantees for Profit, thus we can choose ϵ appropriately to maximize the guaranteed value of Profit. In practice, it turns out we can treat L, the lower bound of all items' efficiency, as a tunable parameter (essentially ignoring all items with efficiency less than L), and significantly improve the performance of the bidding algorithm. We will dicuss this in Section 5.2. The proof of Theorem 4(ii) follows from Theorem 1 setting $L = 1$. The assumption is valid if the budget B is not exceedingly large. In practice, even if the advertiser wants to maximize revenue, rarely is he willing to buy unprofitable keyword positions.

4.1 Multiple-Slot Bidding Strategies

For multiple-slot auctions we consider both profit-maximizing and revenue- maximizing cases. At each time period, bidder 0 has to decide which slot should he outbid to win. The algorithm suggests bidding so as to get maximum profit (revenue) while having a minimum efficiency. Unfortunately, bidding to get maximum profit requires knowledge of other bidders bids. On the other hand, assuming that clickthrough rates increase as we move up the slots, bidding higher would only give a higher revenue.

The profit-maximizing bidding strategy is presented below. The parameters are the same as in the single-slot case. Notice that the bidding strategy is still oblivious of $X(t)$, however now requires knowing the bids $b_s(t)$ and also $\alpha(s)$.

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

$$s = \arg\max\nolimits_{s \in E_t} (V - b_s(t))\alpha(s).$$

For revenue maximization, we can actually find the slot s in time t to maximize the revenue. This is because, the revenue obtained on bidding $b_s(t)$ is $VX(t)\alpha(s)$. Given that $\alpha(s)$ is a decreasing function, maximizing $VX(t)\alpha(s)$ is equivalent to minimize s, i.e., to find the rank s as low as possible. Since the efficiency condition imposes that the slot we win have $b_s(t) \leq \frac{V}{\Psi(z(t))}$, our bid should be exactly that. Thus we have a bidding strategy for revenue-maximizing multiple-slot auctions which is *exactly the same* as that for single-slot auctions and has the desirable property of obliviousness.

Similar to the performance guarantee of the single-slot bidding strategies in Theorem 4, the above bidding strategies have performance guarantees, stated as the following theorem:

Theorem 5. *(i) For multiple-slot profit maximization, for any $\epsilon > 0$,*

$$\mathrm{OPT_p} \leq \epsilon \mathrm{B} + \left(\ln\left(\frac{\mathrm{V}}{\epsilon \mathrm{b_{min}}}\right) + 2\right) \cdot \mathrm{Profit}.$$

(ii) For multiple-slot revenue maximization,

$$\mathrm{OPT_r} \ \leq \ \left(\ln\left(\frac{\mathrm{V}}{\mathrm{b_{min}}}\right) + 2\right) \cdot \mathrm{Revenue}\ .$$

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.

5.1 Simulation and the Sniping Heuristic

We now discuss an experiment for single-slot auctions that points out a weakness of the bidding strategy. We then modify the strategy which, although having the same theoretical guarantee, performs much better empirically. As a negative, the strategy does not remain oblivious any more: it requires knowledge of $X(t)$, the traffic function and also α, the clickthrough-rate of the slot.

Figure 1 shows the performance of our algorithm in a simulation against bidders whose bids are random variables. The budget of the bidder is \$1000 and value $V =$ \$8.00. Figure 1 shows our strategy obtains around 40% of that obtained by the omniscient bidder (the theoretical bound is around 13%). The advertiser stops overbidding very early, at around $t = 200$, and has an unspent budget of \$425.

At time t, suppose the fraction of budget remaining is $y(t) = 1 - z(t)$. Moreover assume we know future click traffic $X(\tau)\alpha$ for $t < \tau \leq T$. Thus the maximum number of clicks in the remaining time is $\int_t^T X(\tau)\alpha \cdot d\tau$, and bidding at most $\frac{y(t)\cdot B}{\int_t^T X(\tau)\alpha \cdot d\tau}$ from time t to T would avoid exhausting the budget. This suggests the following modified strategy which in the toy example of Figure 1 almost doubles the profit.

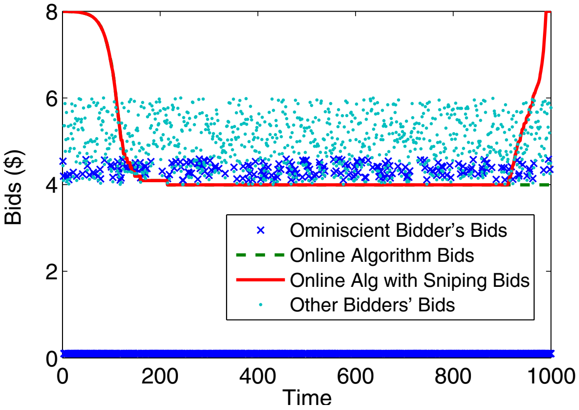

Fig. 1. Performance comparison of various bidding strategies in presence of one other bidder who bids a price uniform random in [4, 6]

Bidding Strategy: Profit-Maximizing Single-Slot with Sniping
Fix $\epsilon > 0$. Let $\Psi(z) \equiv (Ue/\epsilon)^z(\epsilon/e)$.
At time t, if fraction of budget spent is $z(t)$, bid

$$\max\left\{\frac{V}{1+\Psi(z(t))},\ \frac{(1-z(t))\cdot B}{\int_t^T X(\tau)\alpha\cdot d\tau}\right\}.$$

The following theorem shows that the sniping does not affect the worst-case behavior of the strategies.

Theorem 6. *The modified bidding strategy using* sniping *always obtains at least as much profit as the original bidding strategy.*

The above sniping heuristic can be generalized to the multiple-slot case as well.

Bidding Strategy: Multiple-Slot with Sniping
At time t, let $z(t)$ denote fraction of budget spent, $\rho = \Psi(z(t))$
For each slot s, if $\rho > \frac{v_s(t)}{w_s(t)}$ & $b_s(t) \leq \frac{(1-z(t))B}{\alpha(s)\int_t^T X(\tau)d\tau}$:

$\rho = \frac{v_s(t)}{w_s(t)}$

$E_t = \{s \mid \frac{v_s(t)}{w_s(t)}) \geq \rho\}$
bid $b_s(t)$ where $s = \arg\max_{s\in E_t} v_s(t)$

5.2 Evaluation Using Real Bidding Data

Parameter Tunning. 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 $\varPsi$. 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 empirically. If we choose $L = 0.1$ for profit maximization, we get less than 50% performance without sniping and about 70% with sniping. However, with L tuned and fixed for the non-sniping case, we get much better results.

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 continous 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 $b_{\min} = 0.9$. The lower bound L is optimized for each instance without sniping, and it remains the same for the sniping version.

We summarize the experimental results in Table 5.2. For all the examples we run, sniping improves the bidding performance significantly while exhausting the budget. Table 5.2 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/	budget	ALG	ALG/
			OPT	left	(sniping)	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 Concluding Remarks

The algorithms in the paper can be extended to the general case where there are multiple keywords and each keyword has multiple positions. The competitive ratio would now have V replaced by $V_{\max}$, where $V_{\max}$ is the maximum valuation for all keywords. As an open problem, there is a gap of additive constant 1 between the lower and upper bounds for the competitive ratio of Online-MCKP, and it will be nice to close the gap.

References

1. Overture view bids (Febuary 2007), http://uv.bidtool.overture.com/d/search/tools/bidtool
2. Awerbuch, B., Azar, Y., Plotkin, S.A.: Throughput-competitive on-line routing. In: FOCS, pp. 32–40 (1993)
3. Buchbinder, N., Jain, K., Naor, J.S.: Online primal-dual algorithms for maximizing ad-auctions revenue. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 253–264. Springer, Heidelberg (2007)
4. Marchetti-Spaccamela, A., Vercellis, C.: Stochastic on-line knapsack problems. Mathematical Programming 68, 73–104 (1995)
5. Mehta, A., Saberi, A., Vazirani, U.V., Vazirani, V.V.: Adwords and generalized on-line matching. In: Proc. FOCS, pp. 264–273 (2005)
6. Yao, A.C.-C.: Probabilistic computations: towards a unified measure of complexity. In: Proc. 18th IEEE FOCS, pp. 222–227 (1977)