

Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems

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

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

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

ABSTRACT

Sponsored search auctions hosted by major search engines allow advertisers to select relevant keywords, allocate budgets to those terms, and bid on different advertising positions for each keyword in a real-time auction against other advertisers. This dynamic and competitive process creates the significant problem of optimal bid management, especially for large advertisers who need to manage thousands of keywords and spend tens of millions on such advertising. Algorithms for efficient, competitive bid optimization are therefore highly desirable. We approach this problem by casting it as an online (multiple-choice) knapsack problem, and design algorithms for the online knapsack problem achieving a *provably optimal* competitive ratio. This allows for the automation of the bidding process, while optimizing bids to best achieve the goals of the program. To maximize revenue from sponsored search advertising, our bidding strategy can be 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 scraped from the Overture website, and also discuss a sniping heuristic that strictly improves bidding performance.

1. INTRODUCTION

Sponsored search auctions generated an estimated \$10 billion in revenue globally in 2005 and the market is expected to grow 41% in 2006, to more than \$14 billion¹. The results page of a keyword search is apparently an extremely effective place for advertisers to reach an engaged audience. Using an automated auction mechanism, search engines sell the right to place ads next to these keyword results and alleviate the auctioneer from the burden of pricing and placing ads. The intent of the consumer is matched with that of the advertiser through an efficient cost/benefit engine that favors advertisers who offer what consumers seek.

On the advertiser's side, companies (such as HP) spend billions of dollars each year in marketing with an increasingly large portion of that dedicated to search engine marketing. A number of natural questions arise: Game theoretically,

how can an advertiser bid strategically against competitors to maximize relative return? Operationally, how can an advertiser optimize the bidding process assuming that other advertisers have fixed bidding patterns, how to allocate budgets to keywords and how to bid under budget constraints? In this work we focus on the bid optimization question under the budget constraint. Formally, we try to address the following problem: For each keyword and each time period, how much should an advertiser bid to obtain which position, so as to maximize return on investment (ROI) of these auctions?

The bidding strategies we develop are based on the current policy used by search engines to display their ads. We assume that at each query of a keyword, the highest bidder gets the first position, the second highest the second and so on. Moreover, the pricing scheme is the generalized second price scheme [11, 24, 16] where the advertiser in the i -th position pays the bid of the $(i + 1)$ -th advertiser whenever the former's ad is clicked on. For each user click on its ad, the advertiser obtains a *revenue*, which is the expected value-per-click, and a *profit*, which is equal to the difference between revenue and cost. The advertiser (or the agent acting on behalf of the advertiser) has a budget constraint, and would like to maximize either the revenue or the profit. These budget constraints arise out of the ordinary operational constraints of the firm and its interactions with its partners, as well as being a generic feature of keyword auction services themselves.

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. This competitive analysis framework has been used in the worst-case analysis of online algorithms and helps to convert the problem of devising bidding strategies to designing algorithms for online knapsack problems. Although it is known [18] that the most general online knapsack problem admits no online algorithms with any non-trivial competitive ratio, the auction scenario suggests a few constraining assumptions which allow us to give interesting and optimal online algorithms. Thus, we contribute to the literature of knapsack problems as well.

The bidding strategies suggested by the online algorithms are very simple to state. In fact, the bidding strategy for revenue maximization can be stated in one sentence:

At any time t , if the fraction of budget spent is $z(t)$, bid $\frac{V}{\sqrt{z(t)}}$,

where V is the expected value-per-click of the keyword, and

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

¹source: Piper Jaffray & Co.

$\Psi(z)$ is a continuous function of z . Thus the bidding price depends only on the value of the keyword and the fraction of budget spent. Besides its simplicity, the strategy is also *oblivious*, in the sense that it does not need to consider other player’s bids or how frequently queries arrive.

1.1 Model Description

Suppose there are $N + 1$ bidders $\{0, 1, \dots, N\}$ interested in a *single keyword*. Bidder 0 is the default advertiser, and he wants to maximize his profit over a period of time T . Let V denote the expected value-per-click for the default advertiser,² and he has a budget of B over time period T (e.g., if T is 24 hours, B is the daily budget). Here the budget constraint is a *hard* constraint, in the sense that once exhausted, it can not be refilled; budget remaining at the end of period T is taken away. Once a bidder exhausts his budget, he leaves the auction.

Bidders bid on the keyword, and are allowed to change their bids at any moment of time. We make a crucial but reasonable assumption that the bids are very small compared to the budget of Bidder 0. As soon as a query for the keyword arrives, the search engine allocates S slots to bidders as follows: It takes the S highest bids, $b_1 \geq b_2 \geq \dots \geq b_S$ and displays s -th bidder’s ad in slot s . Moreover, if any user clicks on the ad at the s -th slot, the search engine charges the s -th bidder a price b_{s+1} , if $s < S$ or a minimum fee b_{\min} (usually 10¢). Hence, we may assume that all bids are at least b_{\min} .

Each slot s has a click-through-rate $\alpha(s)$, which is defined as the expected number of clicks on an ad divided by the total number of impressions (displays). Usually $\alpha(s)$ is a decreasing function of s . Each time his ad in slot s is clicked, Bidder 0 gets a profit of $V - b_{s+1}$ where b_{s+1} is the bid of the advertiser in the $(s + 1)$ -th slot or b_{\min} if $s = S$. Suppose the time interval T is discretized into periods $\{1, 2, \dots, T\}$, such that, within a single time period t , no bidder changes his bid. Let $X(t)$ denote the expected number of queries for the keyword in time period t . Moreover, suppose Bidder 0 can make his bid in time period t after seeing all other bidders’ bids. As we will see, this assumption does not matter much and is mainly for explanation purposes. The problem faced by Bidder 0 is to decide, how much to bid at each time period t , in order to maximize its profit while keeping its total cost within its budget.

1.2 Bidding Strategies and Knapsack Problems

It is not too hard to see that if we know bids of all the agents at each time period, then the best bidding strategy corresponds to solving a knapsack problem (This observation was also made by [6].) Let us start with the relatively simple single-slot case where there is only one ad slot. At each time period t , let $b(t)$ be the maximum bid on the keyword among bidders 1 to N . The omniscient bidder knows all the bids $\{b(t)\}_{t=1}^T$. To maximize his profit, the omniscient bidder should bid higher than $b(t)$ at those time periods which give him maximum profit and keep his total cost within budget. Winning at time t costs him $w(t) = b(t)X(t)\alpha$ and earns him profit $\pi(t) = (V - b(t))X(t)\alpha$, where $X(t)\alpha$ is the expected number of clicks at time period

²Here we assume that the expected value-per-click for Bidder 0 is fixed over time T .

t . Thus, the omniscient bidder should choose time periods $S \subset T$ to maximize $\pi(S) = \sum_{t \in S} \pi(t)$ satisfying the constraint $w(S) = \sum_{t \in S} w(t) \leq B$. This is a standard instance of the classic *0/1 knapsack problem*, which is defined as following: Given a knapsack of capacity B and T items of profit and weight $(\pi(t), w(t))$ for $1 \leq t \leq T$, select a subset of items to maximize the total profit with total weight of selected items bounded by B . For the case of maximizing revenue, it is similar except that $\pi(t) = VX(t)\alpha$ for each item t . However, for the keyword auction problem, items arrive in an online fashion. At each time period t , Bidder 0 has to make a decision of either overbidding $b(t)$ or not. Bidder 0 does not know the future, and furthermore, it could neither recall time instances gone nor revoke its decision of outbidding later. Thus designing a bidding strategy corresponds to designing an algorithm for the *online* knapsack problem.

The case of multiple slots is captured by the online version of a variant of the classical knapsack problem, the multiple-choice knapsack problem. We elaborate more on this in Section 3.

The knapsack problem is a classic problem in operations research and theoretical computer science. For online knapsack problems, Marchetti-Spaccamela and Vercellis [18] showed that in its most general case, there can be *no* online algorithm achieving any non-trivial competitive ratio, where the *competitive ratio* is the ratio of the value of the given online algorithm to that of the best offline algorithm.³ Fortunately, in our setting, we can make two reasonable assumptions on the items of the knapsack, which allow us to develop interesting online algorithms. We state the assumptions below and justify them in Section 2. The assumptions are:

1. Each item has weight much smaller than the capacity of the knapsack, that is, $w(t) \ll B$ for each item t .
2. The value-to-weight ratio of each item is both lower and upper bounded, i.e., $L \leq \frac{\pi(t)}{w(t)} \leq U$, $\forall t$.

1.3 Our Results

In Section 2, we show that in the case of single-slot auctions, the bidding strategy corresponds to online algorithms for the classical 0/1 knapsack problem. We design an algorithm for the online knapsack problem with competitive ratio $\ln(U/L) + 1$, and also show a lower bound of $\ln(U/L)$. Therefore our algorithm is *provably optimal* in the worst-case sense. We translate the online knapsack algorithm into a bidding strategy for the single-slot auction. As stated in the introduction, these 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.

In Section 3, we extend our results to the case of multiple-slot auctions. We give a $(\ln(U/L) + 2)$ -competitive online algorithm for a variant of the classical knapsack problem called the multiple-choice knapsack problem (MCKP). We translate the algorithm for Online-MCKP to bidding strategies for the multiple-slot case, and obtain both profit-maximizing

³Consider a knapsack of capacity 1 and two sequences $\{(1,1);(0,1)\}$ and $\{(1,1);(\infty,1)\}$. Any deterministic strategy will perform arbitrarily badly against at least one of these sequences.

and revenue-maximizing bidding strategies. The profit maximizing strategy is not oblivious and requires knowledge of other players' bids and also the clickthrough-rates of all slots. The revenue-maximizing strategy remains oblivious.

The reason why the multiple-slot profit-maximizing strategy turns non-oblivious is subtle: It might be more *profitable* for an advertiser to appear in a less desirable (lower) slot and pay less than appearing in a higher slot which gives more clicks. This non-monotonicity has actually been used [11, 3] to show that the generalized second-price scheme is not truthful.

For ease of exposition, we restrict our attention to a single keyword in this paper. All our results extend naturally to the general case of multiple keywords and multiple slots per keyword, with V replaced by V_{\max} , the maximum valuation-per-click among all the keywords.

We implement these bidding strategies and evaluate them using both synthetic bidding data and real bidding data scraped from the Overture website. We modify our strategy by adding a *sniping* heuristic, which while maintaining the same theoretical bounds, performs much better empirically. Our preliminary experimental work, reported in Section 4, suggests that parameter tuning is important for the performance of bidding algorithms.

1.4 Related Work

Keyword Auctions. Keyword auctions are used by Internet search engines to sell various positions (slots) of search results pages of keyword queries. Edelman et al. [11] and Varian [24] study Nash equilibria of a fixed keyword market for strategic bidders without budget constraints. The current auction mechanism deployed by Google and Yahoo! are not truthful, and various truthful auction mechanisms are proposed by Gagan et al. [3] and Lahaie [16]. Bidding dynamics and its impact on auctioneer revenue were discussed by a series of papers [10, 4, 23, 26].

Budget Constraints. When budget constraints are enforced by bidders, things become complicated. For a simplified model where multiple identical units of a single good are auctioned with budget constrained bidders, Borgs et al. [7] and Abrams [2] study the revenue maximization problem with competitive analysis. For the ad placement revenue maximization problem, Mehta et al. [19] propose an online algorithm that assigns queries to bidders achieving an optimal competitive ratio of $1 - 1/e$. The techniques used in this paper are perhaps the closest to the ones we use. They use a trade-off function Ψ (compare it to our threshold function), and grant queries to the bidder having the maximum Ψ value. We discuss more about this similarity in Section 2.1. Despite the similarity in techniques, our work is complementary to their work. While [19] maximize the revenue of the *auctioneer*, we maximize profit or revenue of the advertiser. Thus, while they cast their problem as a generalized online matching between bidders and queries, we cast our problem as a generalized online knapsack problem, and the results of these two papers are incomparable.

Bidding Optimization. For bidding optimization from the advertiser point of view, there has been less work. Kitts and LeBlanc [13] describe various heuristics for bidding optimization in keyword auctions, but do not include any theoretical results. Borgs et al. [6] propose a bidding strategy which over time equalizes the *return on investment* (ROI) over all keywords. They show that in the case of single-

slot auctions, if all bidders play the same strategy, in finite time, the pricing of keywords reach a market equilibrium. Rusmevichientong and Williamson [22] discuss the situation in which an advertiser must learn the clickthrough-rates for various keywords over time and bid accordingly. Their setting does not include any competitive aspect of the bidding process.

Online Knapsack Problems. Our bidding strategies are built on online algorithms for the (multiple-choice) knapsack problems. Marchetti-Spaccamela and Vercellis [18] showed that in the general case, there exists no online algorithm achieving any non-trivial competitive ratio. We make two assumptions in the end of Section 1.1 which are relevant to the keyword auction scenario and make the problem easier to handle. Many special cases of the problem have been studied, including the stochastic online knapsack problem [17, 21, 15], the removable online knapsack problem [12] and the online partially fractional knapsack problems [20], but none of them seem to imply our assumptions. A special case of the online knapsack problem where all items have unit cost is the so-called online multiple secretary problem: numbers come in an online fashion and the task is to select at most k of them to maximize the sum of selected numbers. Kleinberg [14] gave an online algorithm with competitive ration $1 - O(1/\sqrt{k})$ for the online k -secretary problem.

Online Routing and Packing Problems. The online knapsack problem is closely related to other online problems, especially the online admission control and routing problem. Awerbuch et al. [5] studied the online call routing problem where call requests arrive for a single source-destination pair and bandwidth is limited in the network. This is a generalization of the classical knapsack problem (where the network consists of two nodes and one edge). Their results with extensions that assign "profits" to calls seem to imply an $O(\ln(U/L))$ -competitive algorithm for the classic 0/1 knapsack problem. Recently, Buchbinder and Naor [8] came up with a general paradigm for designing online algorithms for the *fractional* packing-covering problems. For the online fractional packing problem with just 1 linear constraint, their work implies an $O(\log(U/L))$ -competitive online algorithm for the knapsack problem where each item can be taken with an arbitrary nonnegative quantity. They also gave a matching $\Omega(\log(U/L))$ lower bound. It is possible to obtain the same competitive ratio enforcing 0/1 quantity constraints for knapsack items combining randomized rounding and other advanced techniques. In a later paper [9], they showed that results stated above in [5] can be derived via their general method.

Although the online knapsack problems (both the 0/1 and the multiple-choice) fall into the online routing and packing framework, it is not clear how to enforce the 0/1 quantity constraint to obtain an $O(\log(U/L))$ -competitive algorithm, and how to enforce the xor constraint for the multiple-choice knapsack problem. Furthermore, our bounds for the competitive ratio are upper bounded by $\ln(U/L) + 1$ and lower bounded by $\ln(U/L)$, respectively. One can think of our results as pinning down the constants in their results for the knapsack problem via a simple algorithm. Furthermore, their settings seem not generalize to the multiple-choice knapsack problem, which is actually the most important case for the keyword auction model. In addition, our algorithms are much more direct and cleaner, and thus give

rise to simple and oblivious bidding strategies.

As an anonymous reviewer pointed out, a simple randomized algorithm implies a competitive ratio of $O(\log(U/L))$ for the online knapsack problem. The algorithm first picks uniformly random an integer i from $[0, \log(U/L)]$ and set the threshold value $x = 2^i$. For each arriving item, it will be selected if and only if the knapsack is not full and the item efficiency ratio is at least x . The algorithm may return a profit of 0, has a high variance of returned value, and is no longer competitive if the adversary can change input stream after seeing the threshold value. On the other hand, our algorithms are robust to any adversary attack, and the worst case bound $\ln(U/L)$ holds for each run of any instance.

2. SINGLE-SLOT AUCTIONS AND ONLINE KNAPSACK PROBLEMS

In this section, we focus on single-slot auctions and the corresponding online knapsack problem. We design algorithms for the online knapsack problem and translate them back into bidding strategies for single-slot keyword auctions. Before presenting the algorithms, we first explain why the assumptions made in Section 1.2 are justified.

Recall that the unique item at time period t has a weight $w(t)$ and a profit $\pi(t)$ where

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

For revenue maximization, $\pi(t)$ corresponds to the revenue of winning the bid, thus $\pi(t) = VX(t)\alpha$. Here $X(t)\alpha$ is the number of expected clicks on the displayed ad in time period t . The first assumption of $w(t) \ll B$ follows since the budget of the agent is usually much larger than the money spent in small time periods as the bids are small. For the second assumption, we separate into two cases. In the case of profit maximization, since $\frac{\pi(t)}{w(t)} = \frac{V}{b(t)} - 1$, it suffices to set $U \equiv \frac{V}{b_{\min}} - 1$. To get a lower bound on the profit-to-weight ratio, notice that if $b(t)$ is close to V , little is lost without bidding at those time intervals. Specifically, if we bid only when $b(t) \leq \frac{V}{1+\epsilon}$ for some fixed $\epsilon > 0$, the maximum amount of profit lost from not bidding in these time periods is bounded by ϵB . If ϵ is small, then the profit loss can be negligible. In other words, we can set $L = \epsilon$ and ignore all items with efficiency smaller than ϵ . This results in a maximum profit loss of ϵB , and it will not affect much the algorithm performance if the total profit of the algorithm is relatively large. In the case of revenue maximization, we have $\pi(t)/w(t) = V/b(t)$. Here it suffices to set $U \equiv \frac{V}{b_{\min}}$. For the lower bound with revenue maximization, it is reasonable to assume that *the optimum strategy would never bid when $b(t)$ is higher than V* . This holds when there are enough items with value-to-cost ratio at least 1 to consume all the budget. Otherwise, the optimal solution needs to take items which cost more than their value, and the budget seems unnecessarily large. Therefore, assuming that the optimum never bids when $b(t)$ is higher than V , we only need to consider items with efficiency at least 1, i.e., set $L = 1$.

2.1 The Online Knapsack Problem

Given an online algorithm \mathcal{A} , we say that \mathcal{A} is c -competitive (has a competitive ratio of c) if for any input sequence of items σ , we have $\text{OPT}(\sigma) \leq c\mathcal{A}(\sigma)$, where $\mathcal{A}(\sigma)$ is the profit

of \mathcal{A} given σ , and $\text{OPT}(\sigma)$ is the maximum profit obtained by any *offline* algorithm with the knowledge of σ .

We give an algorithm which makes the assumptions of Section 1.2, that is for all items t , $w(t) \ll B$ and $L \leq \frac{\pi(t)}{w(t)} \leq U$. Given the lower bound L and upper bound U for item efficiency, it is easy to verify that the greedy algorithm which keeps selecting items until the knapsack is full gives a competitive ratio of U/L . Next we state our algorithm for the online knapsack problem, which achieves an almost 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

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

Observe that for $z \in [0, \bar{z}]$ where $\bar{z} \equiv 1/\ln(Ue/L)$, $\Psi(z) \leq L$, thus the algorithm will pick all items available until \bar{z} fraction of the knapsack is filled. When $z = 1$, $\Psi(z) = U$, and since Ψ is strictly increasing, the algorithm will not spend more than its budget.

The above algorithm may seem mysterious to the reader as it uses just one threshold function to select items, and the threshold function is a very specialized exponential function of its capacity filled. Actually we first consider a discrete version which in the limit leads us to the continuous threshold function Ψ , for details see the forthcoming technical report. This method of obtaining the continuous version of a discrete algorithm was also done in [19].

THEOREM 2.1. *For any input sequence σ , if $\mathcal{A}(\sigma)$ is the profit obtained by ONLINE-KP-THRESHOLD and $\text{OPT}(\sigma)$ is the maximum profit that can be attained, then*

$$\text{OPT}(\sigma) \leq \mathcal{A}(\sigma)(\ln(U/L) + 1).$$

In other words, the above algorithm has a competitive ratio of $\ln(U/L) + 1$.

PROOF. Fix the input sequence σ . Let the algorithm terminate filling Z fraction of the knapsack and obtaining a profit 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 profit of the common items by $W = w(S \cap S^*)$ and $P = \pi(S \cap S^*)$. 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,

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

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

$$\frac{\text{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{P + \Psi(Z)(B - W)}{P + \pi(S \setminus S^*)}. \quad (1)$$

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. Round down the profit of each item j picked by the algorithm to $\Psi(z_j)w_j$ where w_j is weight of the item. Let P_1 be

the profit obtained by rounding down profits of $S \cap S^*$ and P_2 be profit obtained by rounding down $S \setminus S^*$, then

$$P \geq P_1 \equiv \sum_{j \in S \cap S^*} \Psi(z_j)w_j, \quad (2)$$

$$\pi(S \setminus S^*) \geq P_2 \equiv \sum_{j \in S \setminus S^*} \Psi(z_j)w_j. \quad (3)$$

If $\Psi(Z)(B - W) \leq \pi(S \setminus S^*)$, then Eq. (1) implies that $\text{OPT}(\sigma) \leq \mathcal{A}(\sigma)$, thus the claim of the theorem is trivially satisfied. Therefore we now assume that $\Psi(Z)(B - W) > \pi(S \setminus S^*)$. Thus, if we reduce P to P_1 in both denominator and numerator in the RHS of Eq. (1), the ratio increases, i.e.,

$$\frac{P + \Psi(Z)(B - W)}{P + \pi(S \setminus S^*)} \leq \frac{P_1 + \Psi(Z)(B - W)}{P_1 + \pi(S \setminus S^*)} \quad (4)$$

Notice that $P_1 \leq \Psi(Z)w(S \cap S^*) = \Psi(Z)W$, since Ψ is an increasing function, thus

$$\frac{P_1 + \Psi(Z)(B - W)}{P_1 + \pi(S \setminus S^*)} \leq \frac{\Psi(Z)B}{P_1 + P_2} \quad (5)$$

Combining Eqs.(1), (4), (5), we get

$$\frac{\text{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{\Psi(Z)B}{P_1 + P_2} = \frac{\Psi(Z)}{\sum_{j \in S} \Psi(z_j)\Delta z_j} \quad (6)$$

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

Based on Assumption 1 in Section 1.2, $\Delta z_j \approx 0$ (Refer to the remark following the proof), and thus

$$\sum_{j \in S} \Psi(z_j)\Delta z_j \approx \int_0^Z \Psi(z)dz.$$

Note that since $\Psi(z) \leq L$ for $0 \leq z \leq \bar{z}$ where $\bar{z} \equiv 1/\ln(Ue/L)$, we can replace $\Psi(z)$ by L in this interval. Thus,

$$\begin{aligned} \frac{P_1 + P_2}{B} &\geq \int_0^{\bar{z}} L \cdot dz + \int_{\bar{z}}^Z \Psi(z) \cdot dz \\ &= L\bar{z} + \frac{L}{e} \frac{(Ue/L)^Z - (Ue/L)^{\bar{z}}}{\ln(Ue/L)} \\ &= \frac{L}{e} \frac{(Ue/L)^Z}{\ln(Ue/L)} = \frac{\Psi(Z)}{\ln(Ue/L)} \end{aligned}$$

Putting this together with Eq.(6), we complete the proof of Theorem 2.1.

$$\frac{\text{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{\Psi(Z)B}{P_1 + P_2} \leq \ln(Ue/L) = \ln(U/L) + 1.$$

□

REMARK 2.1. 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.

2.2 Lower Bound on Competitive Ratio of Online Knapsack Problem

In this section we use Yao's minimax technique [25] to get a lower bound on the competitive ratio of Online-KP, almost matching the upper bound given in Theorem 2.1.

That is, we show if any deterministic (or randomized) algorithm is c -competitive for the online knapsack problem, i.e., for any input sequence σ it guarantees $\text{OPT}(\sigma) \leq c\mathcal{A}(\sigma)$ (or $\text{OPT}(\sigma) \leq c\mathbf{E}[\mathcal{A}(\sigma)]$), then $c \geq \ln(U/L)$. We state our results in the following theorem:

THEOREM 2.2. *With the assumptions stated in Section 1.2, the competitive ratio of any (possibly randomized) online algorithm is at least $\ln(U/L)$.*

PROOF. Yao's minimax principle gives us the following relation. For any input distribution D and any randomized algorithm \mathcal{A} ,

$$\min_{\sigma} \frac{\mathbf{E}[\mathcal{A}(\sigma)]}{\text{OPT}(\sigma)} \leq \max_{\text{deterministic } A} \mathbf{E}_{\sigma \leftarrow D} \left[\frac{A(\sigma)}{\text{OPT}(\sigma)} \right]$$

and thus the right-hand side gives an upper bound on $1/c$. Note that the first expectation is over the random choices that the randomized algorithm makes, and the second expectation is over the randomization in the input distribution. Thus to prove the lower bound, we specify a distribution D such that

$$\max_{\text{deterministic } A} \mathbf{E}_{\sigma \leftarrow D} \left[\frac{A(\sigma)}{\text{OPT}(\sigma)} \right] \leq \frac{1}{\ln(U/L)}. \quad (7)$$

Fix a parameter $\eta > 0$. Let k be an integer such that

$$(1 + \eta)^k = U/L, \quad \text{i.e., } k = \frac{\ln(U/L)}{\ln(1 + \eta)}.$$

The support of the input distribution consists of the instances I_0, I_1, \dots, 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 D is specified by giving equal probability of $\frac{1}{k+1}$ to each of these instances.

Given this distribution, any deterministic algorithm A can be specified by the vector (f_0, f_1, \dots, f_k) , where f_i is the fraction of the knapsack it fills with items having value-to-weight ratio $(1 + \eta)^i L$. Thus we have

$$\begin{aligned} &\mathbf{E}_{\sigma \leftarrow D} \left[\frac{A(\sigma)}{\text{OPT}(\sigma)} \right] \\ &= \frac{1}{k+1} \sum_{i=0}^k \frac{\sum_{j=0}^i (1 + \eta)^j f_j}{(1 + \eta)^i} \\ &= \frac{1}{k+1} \sum_{j=0}^k f_j \sum_{i=j}^k (1 + \eta)^{j-i} \\ &\leq \frac{1}{k+1} \sum_{j=0}^k f_j \frac{1 + \eta}{\eta} \\ &\leq \frac{1 + \eta}{k\eta} = \frac{(1 + \eta) \ln(1 + \eta)}{\ln(U/L)\eta}, \end{aligned}$$

where the last inequality uses the fact that $\sum_{i=0}^k f_i \leq 1$ as the algorithm cannot over-fill the knapsack.

Let $\eta \rightarrow 0$, then $\ln(1 + \eta)/\eta \rightarrow 1$, and the last term of the above inequalities goes to $1/\ln(U/L)$. So Eq.(7) is proved by setting η approaching 0. □

2.3 Bidding Strategies for Single-Slot Auctions

We now construct the bidding strategies suggested by the algorithm ONLINE-KP-THRESHOLD. Let us consider the profit-maximizing case first. As explained in the beginning of the section, we may assume $b(t) \leq \frac{V}{1+\epsilon}$ for some ϵ . Set $U = V/b_{\min} - 1$ and $L = \epsilon$. At time t , suppose Bidder 0 has spent $z(t)$ fraction of its budget. This corresponds to the fact that $z(t)$ fraction of the knapsack is filled. As per the algorithm, Bidder 0 must win the bid if and only if the efficiency of the next bid is at least $\Psi(z(t))$. That is, Bidder 0 must bid higher than $b(t)$ iff $\frac{V-b(t)}{b(t)} \geq \Psi(z(t))$ or $b(t) \leq \frac{V}{1+\Psi(z(t))}$, which means bidding $\frac{V}{1+\Psi(z(t))}$ suffices. We formally state the bidding strategy below:

Bidding Strategy: PROFIT-MAXIMIZING SINGLE-SLOT
 Fix $\epsilon > 0$. Let $\Psi(z) \equiv (Ue/\epsilon)^z(\epsilon/e)$.
 At time t , if fraction of budget spent is $z(t)$, then bid

$$b_0(t) = \frac{V}{1 + \Psi(z(t))}.$$

The above bidding strategy for profit-maximizing single-slot auctions has the following performance guarantee:

THEOREM 2.3. *Let Profit denote the profit obtained by our profit-maximizing bidding strategy. Then for any $\epsilon > 0$,*

$$\text{OPT} \leq \epsilon B + \ln\left(\frac{e(V - b_{\min})}{\epsilon b_{\min}}\right) \cdot \text{Profit}$$

where b_{\min} is the minimum bid of any bidder and OPT is the maximum profit obtained by the omniscient bidder.

The proof of Theorem 2.3 follows from Theorem 2.1 and the fact that the optimum without the assumption of $b(t) \leq \frac{V}{1+\epsilon}$ can actually earn at most ϵB more profit than the optimum with the assumption.

Similarly, using $U = \frac{V}{b_{\min}}$ and $L = 1$, we get the following revenue-maximizing bidding strategy.

Bidding Strategy: REVENUE-MAXIMIZING SINGLE-SLOT
 At time t , if fraction of budget spent is $z(t)$, then bid

$$b_0(t) = \frac{V}{\Psi(z(t))}$$

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

THEOREM 2.4. *Let Revenue be the profit obtained by our revenue-maximizing bidding strategy and OPT be the maximum revenue obtained by the omniscient bidder. Assuming that OPT does not contain any item t with $b(t) > V$, then*

$$\text{OPT} \leq \ln\left(\frac{eV}{b_{\min}}\right) \cdot \text{Revenue}$$

where b_{\min} is the minimum bid of any bidder.

The proof of Theorem 2.4 follows from Theorem 2.1. Note that both the above strategies looked only at the budget spent and are thus oblivious of other bidders' bids and the number of clicks for each time slot.

3. MULTIPLE-SLOT AUCTIONS AND ON-LINE MCKP

We extend our investigation to the case of multiple slots. As we see, the strategy in the multiple-slot case corresponds to the online multiple-choice knapsack problem. The multiple-choice knapsack problem (MCKP) is a generalization of the knapsack problem: Given a knapsack of capacity B , and T sets of items N_1, N_2, \dots, N_T , the goal is to choose at most one item from each set to maximize profit and not exceed the knapsack capacity. In the online version, the sets come online and we need to make our decision on the fly. Items once picked cannot be disposed, and sets cannot be recalled.

Next we show how to model the multiple-slot keyword auction problem into Online MCKP. Once again, the budget corresponds to the capacity of the knapsack. At each time period t , let $b_1(t) \geq b_2(t) \geq \dots \geq b_S(t)$ be the S highest bids. To win slot s , we need to bid $b_s(t)$. This incurs a cost $w_s(t)$ and a profit $\pi_s(t)$ where

$$w_s(t) \equiv b_s(t)X(t)\alpha(s), \quad \pi_s(t) \equiv (V - b_s(t))X(t)\alpha(s),$$

and $\alpha(s)$ is the click-through-rate of slot s . The S slots at each time period correspond to the set arriving at time t . Since Bidder 0 can win at most one slot at each time period, the omniscient bidder needs to solve the multiple-choice knapsack problem while Bidder 0 needs to solve the online multiple-choice knapsack problem. Once again, we assume (1) $w_s(t) \ll B$ and (2) $L \leq \pi_s(t)/w_s(t) \leq U$ for all s, t , similar to the assumptions made in Section 1.2. In subsequent subsections, we give competitive algorithms for Online-MCKP and use it to devise bidding strategies for multiple-slot auctions. The algorithm is an easy generalization of the algorithm ONLINE-KP-THRESHOLD. However, unlike the single-slot case, the bidding strategy for profit maximization requires knowledge of other bidder's bids and also the clickthrough-rates of various slots.

3.1 Algorithm for Online MCKP

The algorithm for Online-MCKP is very similar to that for Online-KP, which is stated below.

Algorithm ONLINE-MCKP-THRESHOLD
 Let $\Psi(z) \equiv (Ue/L)^z(L/e)$.
 At time t , let $z(t)$ denote the fraction of capacity filled,

$$E_t \equiv \left\{ s \in B_t \mid \frac{\pi_s(t)}{w_s(t)} \geq \Psi(z(t)) \right\},$$

pick element $s \in E_t$ with maximum $\pi_s(t)$

The above algorithm has a competitive ratio of $\ln(U/L) + 2$, stated as the following theorem:

THEOREM 3.1. *The algorithm Online-MCKP-Threshold for Online-MCKP satisfies*

$$\text{OPT} \leq (\ln(U/L) + 2)\text{ALG}$$

where ALG is the profit of the algorithm, and OPT is the maximum profit of any offline algorithm.

PROOF. For any input sequence of sets σ , let $\mathcal{A}(\sigma)$ be the profit obtained by the above algorithm and $\text{OPT}(\sigma)$ be the maximum profit obtainable. We claim that for any σ ,

$$\text{OPT}(\sigma) - \mathcal{A}(\sigma) \leq (\ln(U/L) + 1)\mathcal{A}(\sigma).$$

If the claim is true, immediately we obtain that

$$\text{OPT}(\sigma) \leq (\ln(U/L) + 2) \mathcal{A}(\sigma)$$

for any input sequence σ , and the theorem is proved. In the following we show how to prove the above claim.

As in the proof of Theorem 2.1, let S and S^* be the set of items picked by the algorithm and the optimum, respectively. Let $P = \pi(S \cap S^*)$ denote the profit of the common items, $W = w(S \cap S^*)$ denote the weight. As before, we want to bound the profit of the items picked by OPT but not by ALG. In the multiple-choice case, unlike in the proof of Theorem 2.1, the efficiency of an item selected by OPT from N_t is not necessarily bounded by $\Psi(z(t))$ since ALG may have also selected one different item from N_t . Thus we partition the items picked by OPT and not by ALG into two: items which do not satisfy the efficiency condition, and the items which do. Thus the first kind of items have efficiency less than $\Psi(z(t))$, while for the second kind of items, the total profit of these items is less than $\mathcal{A}(\sigma)$ since ALG picks the most profitable item from the same set which satisfy the efficiency condition. We can exclude the second types of items from further consideration since they in total result in at most a profit of $\mathcal{A}(\sigma)$. Now we can assume that all items have efficiency $< \Psi(z(t))$ at time t , thus it returns to a similar situation as in the proof of Theorem 2.1. A similar proof shows that the above claim holds. \square

3.2 Bidding Strategy for Multiple-Slot Auctions

For multiple-slot auctions, we also consider the profit-maximizing case first. As in the single-slot case, we assume that $b_s(t) \leq \frac{V}{1+\epsilon}$ for all s, t . This implies that the efficiency of each bid is upper bounded by $U = \frac{V}{b_{\min}} - 1$ and lower bounded by ϵ . So we obtain the following bidding strategy:

Bidding Strategy PROFIT-MAXIMIZING MULTIPLE-SLOT
 Fix $\epsilon > 0$. Let $\Psi(z) \equiv (U\epsilon/\epsilon)^z(\epsilon/\epsilon)$.
 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

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

Note that the bidding strategy is still oblivious of $X(t)$, however now requires knowing the bids $b_s(t)$ and also $\alpha(s)$. Similar to the performance guarantee of the single-slot profit-maximizing bidding strategy in Theorem 2.3, the above bidding strategy has a performance guarantee, stated as the following theorem:

THEOREM 3.2. *Let Profit denote the profit obtained by our profit-maximizing bidding strategy. Then for any $\epsilon > 0$,*

$$\text{OPT} \leq \epsilon B + \left(\ln \left(\frac{V}{\epsilon b_{\min}} \right) + 2 \right) \text{Profit}$$

where b_{\min} is the minimum bid of any bidder and OPT is the maximum profit obtained by the omniscient bidder.

For revenue maximization, the bidding strategy is similar to profit maximization except that 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 in Section 2, which has the desirable property of obliviousness.

THEOREM 3.3. *Let Revenue be the revenue obtained by our revenue-maximizing bidding strategy and OPT the maximum revenue obtained by the omniscient bidder. Assuming that OPT does not contain any item with $b_s(t) > V$, then*

$$\text{OPT} \leq (\ln(V/b_{\min}) + 2) \text{Revenue}$$

where b_{\min} is the minimum bid of all bidders.

4. EXPERIMENTAL EXPLORATION

In this section we conduct preliminary experiments evaluating our bidding algorithms for both synthetic data as well as some limited real-world bidding data.

4.1 Simulation and Strategy Modification

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.

The reason for the above weakness is that the strategy is unaware of the time remaining in the auction. It stops overbidding too early, missing out possible advantageous bids later on. Thus a potential performance improvement is *sniping* towards the end of the auction. 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. Therefore the modified bidding strategy is as follows.

Bidding Strategy: PROFIT-MAXIMIZING SINGLE-SLOT WITH SNIPING
 Fix $\epsilon > 0$. Let $\Psi(z) \equiv (U\epsilon/\epsilon)^z(\epsilon/\epsilon)$.
 At time t , if fraction of budget spent is $z(t)$, bid

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

On running the bidding strategy on the simulation (Refer Figure 1) we found that the strategy attained more than 70% of that of the omniscient bidder and ended up with unspent budget of around \$50.

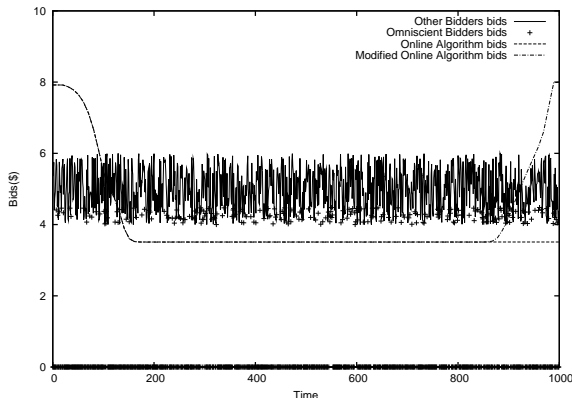


Figure 1: Performance comparison of various bidding strategies in presence of random bidders. Each bidder bids a price uniform random between \$4.00 and \$6.00. Modified Strategy attains around 70% of that obtained by Omniscient Bidder. It emulates original strategy up to time slot 800.

Thus it does much better than the original strategy, although it requires the knowledge of $X(t)$ and α . In fact, as we prove next, the modified bidding strategy with sniping always performs at least as well as the original strategy, implying that Theorem 2.3 holds for the modified bidding strategy as well.

THEOREM 4.1. *The modified bidding strategy using sniping always obtains at least as much profit as the original bidding strategy.*

PROOF. We proof the theorem by showing that whenever the original strategy wins a bid, the modified strategy also wins. Let $p_1(t)$ denote the first term of the modified bid function, and $p_2(t)$ denote the second term of the modified bid function. Since $\Psi(z(t))$ is monotone increasing in term of time t , $p_1(t) = V/(1 + \Psi(z(t)))$ is monotone decreasing in t . Consider the first time t_0 when $p_2(t_0) > p_1(t_0)$. If such t_0 does not exist, the modified bidding strategy is identical to the original bidding strategy and the theorem is trivially proved. Thus we assume that $t_0 \leq T$ exists. Next we claim that $p_2(t)$ is monotone increasing for all $t \geq t_0$. If this is true, then since $p_2(t)$ monotone increasing and $p_1(t)$ monotone decreasing, thus $p_2(t) \geq p_1(t)$ for all $t \geq t_0$.

For simplicity, let $\alpha = 1$. Denote the second term

$$p_2(t) \equiv \frac{(1 - z(t))B}{\int_t^T X(\tau)d(\tau)} = \frac{F(t)}{G(t)}$$

where $F(t) \equiv (1 - z(t))B$, $G(t) \equiv \int_t^T X(\tau)d(\tau)$.

Next we prove that $p_2(t+1) > p_2(t)$. Notice that $F(t+1)$ is budget remaining at time $t+1$, thus it is equal to budget remaining at time t , $F(t)$, minus money spent at time t . Since money spent at time t is at most $X(t) \max\{p_1(t), p_2(t)\} = X(t)p_2(t)$, thus

$$F(t+1) \geq F(t) - X(t)p_2(t) = F(t) \left(1 - \frac{X(t)}{G(t)}\right).$$

Since

$$G(t+1) = G(t) - X(t) = G(t) \left(1 - \frac{X(t)}{G(t)}\right),$$

thus

$$p_2(t+1) = \frac{G(t+1)}{F(t+1)} \geq \frac{G(t)}{F(t)} = p_2(t).$$

Since $p_1(t)$ is monotone decreasing in t , and $p_2(t)$ is monotone increasing when $t \geq t_0$, thus the modified bidding strategy coincides with the original strategy up to time t_0 and then switches to the sniping strategy. Since the sniping strategy is defined to never exceed the budget, the modified bidding strategy never exceeds its budget. \square

The above sniping heuristic can be generalized to the multiple-slot case and it is formally described below. Notice that the theoretical performance bound of Theorem 4.1 does not generalize here, even though we will show in the next subsection that the general sniping heuristic improves bidding performance significantly and exhausts budgets in most experiments.

Bidding Strategy: MULTIPLE-SLOT WITH SNIPIING

At time t , let $z(t)$ denote fraction of budget spent, $\rho = \Psi(z(t))$

For each slot s , if $\rho > \frac{\pi_s(t)}{w_s(t)}$ & $b_s(t) \leq \frac{(1-z(t))B}{\alpha(s) \int_t^T X(\tau)d\tau}$:

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

$$E_t = \{s \mid \frac{\pi_s(t)}{w_s(t)} \geq \rho\}$$

bid $b_s(t)$ where $s = \arg \max_{s \in E_t} \pi_s(t)$

4.2 Evaluation using Real Bidding Data

Next we report some preliminary experimental results on evaluating bidding algorithms for multiple-slot auctions using real bidding data. Due to lack of publicly available bidding data set, we *manually* collected bidding prices associated with each position from the Overture webpage [1]. We launched an Internet Explorer browser visiting Overture view bids website, and periodically refreshed the webpage and downloaded bidding data from it. Due to the website's anti-crawling policy, we had to periodically answer randomly generated computer tests to keep the crawling process alive. Given all the obstacles, we still managed to download data for a couple of weeks, for one of the most dynamic keyword "auto insurance". Each time slot is roughly 1 minute, since we found price changes for this keyword for roughly every one minute. For $T = 1842$, $B = \$1000$, and three different values $V = 8.0, 10.0, 12.0$, we evaluated both profit-maximizing and revenue-maximizing bidding algorithms, with and without sniping.⁴

We observe that the performance of these algorithms depend crucially on L, U , the minimum and maximum efficiency ratios of all the slots. If we use $U = V/b_{\min}$ and $V = 8.0, b_{\min} = 0.1, L = 0.1$, we get a performance ratio of 27%, close to the worst-case guarantee $1/\ln(eU/L)$. On

⁴We also need to know $X(t)$ and $\alpha(s)$ for comparison purposes. For simplicity, we set $X(t) = 1$ and $\alpha(s) = 1 - sd$ for a small constant d for all the experiments. Other reasonable values of $X(t)$ and $\alpha(s)$ lead to similar results and omitted.

the other hand, if we use $b_{\min} = 0.9$ as it is a better empirically observed lower bound for almost all time periods for this keyword, and $L = 4.5$, ALG/OPT turns out to be 53%, with budget remaining \$225. With sniping deployed, the performance improves to 68%, and budget remaining becomes almost zero.

Since results are very similar for different parameter values, we summarize them in Table 1. 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. For all the examples we run, sniping improves the bidding performance significantly while exhausting the budget. The preliminary results in Table 1 seems to tell us, for almost all values, the performance ratio (ALG/OPT) is around 50% without sniping, and 67% with sniping.

Profit-Maximization Bidding Algorithm				
V	OPT	ALG ALG/OPT	budget left	ALG ALG/OPT (sniping)
8.0	5172.5	2751.3 53.2%	225.5	3540.9 68.5%
10.0	6894.4	4058.6 58.9%	116.1	4606.6 66.8%
12.0	8616.2	4462.9 51.8%	240.8	5842.1 67.8%
Revenue-Maximization Bidding Algorithm				
8.0	6887.4	3627 52.7%	194.7	4505.4 65.4%
10.0	8609.3	4234.5 49.2%	236.4	5565 64.6%
12.0	10331.1	5080.8 49.2%	239.8	6700.8 64.9%

Table 1: Performance on “Auto Insurance” for both profit and revenue maximizations.

5. EXTENSIONS AND OPEN QUESTIONS

In this paper we investigated budget constrained bidding strategies in keyword auctions and converted them to variants of online knapsack problems. We focused on the single keyword case, and looked at strategies for both single-slot and multiple-slot cases. We compared performance of our strategies with that of an omniscient bidder who knows every bidder’s bid in advance. This competitive analysis framework translated the bidding problem to online algorithms for (multiple-choice) knapsack problems. In the case of single-slot auctions, our algorithms suggest oblivious strategies for both profit and revenue maximizations. The obliviousness is a very desirable property in any bidding strategy. In the multiple-slot case, our bidding strategy for profit maximization is no longer oblivious and requires knowledge of the various bids and click-through-rates of slots. However, for revenue maximization, it still remains oblivious.

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.

We use worst-case competitive analysis, comparing our bidding strategy with the omniscient bidder who know every-

thing in advance. In practice, other bidders do not behave in the worst-case but bid according to their own strategies. It would be interesting if one could attain a better bidding strategy with some assumptions on the strategies used by other agents. Incorporating previous work on stochastic knapsack problems together with average-case analysis (e.g. Lueker [17]) might be an essential ingredient.

There is a small gap of 1 in the lower and upper bounds for the competitive ratio of the online knapsack problem, and the gap becomes 2 for the multiple-choice case. It will be nice to close these gaps from a theoretical point of view.

Acknowledgement: We thank Gagan Aggarwal, Terence Kelly, Paul O’Brien, David Pennock and Jim Saxe for helpful comments.

6. REFERENCES

- [1] Overture view bids. <http://uv.bidtool.overture.com/d/search/tools/bidtool>.
- [2] Z. Abrams. Revenue maximization when bidders have budgets. In *Proc. SODA*, pages 1074–1082, 2006.
- [3] G. Aggarwal, A. Goel, and R. Motwani. Truthful auctions for pricing search keywords. In *Proc. 7th ACM Conference on Electronic Commerce*, pages 1–7, Ann Arbor, MI, June 2006.
- [4] K. Asdemir. Dynamics of bidding in search engine auctions: An analytical investigation. In *Proc. 2nd Workshop on Sponsored Search Auctions*, 2006.
- [5] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *Proc. of 34th FOCS*, pages 32-40, 1993.
- [6] C. Borgs, J. Chayes, O. Etesami, N. Immorlica, K. Jain, and M. Mahdian. Bid optimization in online advertisement auctions. In *Proc. 2nd Workshop on Sponsored Search Auctions*, Ann Arbor, MI, 2006.
- [7] C. Borgs, J. T. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *Proc. 6th ACM Conference on Electronic Commerce*, pages 44–51, 2005.
- [8] N. Buchbinder and J. Naor. Online Primal-Dual algorithms for Covering and Packing problems. In *Proc. ESA 2005*
- [9] N. Buchbinder and J. Naor. Improved bounds for online routing and packing via a primal-dual approach. In *Proc. FOCS 2006*
- [10] B. Edelman and M. Ostrovsky. Strategic bidder behavior in sponsored search auctions. In *Proc. 1st Workshop on Sponsored Search Auctions*, Vancouver, Canada, June 2005.
- [11] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. In *Proc. 2nd Workshop on Sponsored Search Auctions*, 2006.
- [12] K. Iwama and S. Taketomi. Removable online knapsack problems. In *ICALP*, pages 293–305, 2002.
- [13] B. Kitts and B. Leblanc. Optimal bidding on keyword auctions. *Electronic Markets*, 14(3):186–201, 2004.
- [14] R.D. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. *Proc. SODA*, pages 630-631, 2005.
- [15] A. J. Kleywegt and J. D. Papastavrou. The dynamic

- and stochastic knapsack problem. *Operations Research*, 46(1):17–35, 1998.
- [16] S. Lahaie. An analysis of alternative slot auction designs for sponsored search. In *Proc. 7th ACM Conference on Electronic Commerce*, pages 218–227, Ann Arbor, MI, June 2006.
- [17] G. S. Lueker. Average-case analysis of off-line and on-line knapsack problems. *Journal of Algorithms*, 29(2):277–305, 1998.
- [18] A. Marchetti-Spaccamela and C. Vercellis. Stochastic on-line knapsack problems. *Mathematical Programming*, 68:73–104, 1995.
- [19] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. Adwords and generalized on-line matching. In *Proc. FOCS*, pages 264–273, 2005.
- [20] J. Noga and V. Sarbua. An online partially fractional knapsack problem. In *Proc. 8th Sym. Parallel Architectures, Algorithms and Networks*, pages 108–112, 2005.
- [21] J. D. Papastavrou, S. Rajagopalan, and A. J. Kleywegt. The dynamic and stochastic knapsack problem with deadlines. *Management Science*, 42(12):1706–1718, 1996.
- [22] P. Rusmevichientong and D. P. Williamson. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In *Proc. 7th ACM conference on Electronic commerce*, pages 260–269, 2006.
- [23] B. K. Szymanski and J.-S. Lee. Impact of ROI on bidding and revenue in sponsored search advertisement auctions. In *Proc. 2nd Workshop on Sponsored Search Auctions*, Ann Arbor, MI, 2006.
- [24] H. R. Varian. Position auctions. *International Journal of Industrial Organization*, 2006, to appear.
- [25] A. C.-C. Yao. Probabilistic computations: towards a unified measure of complexity. In *Proc. 18th IEEE FOCS*, pages 222–227, 1977.
- [26] Y. Zhou and R. Lukose. Vindictive bidding in keyword auctions. In *Proc. 2nd Workshop on Sponsored Search Auctions*, Ann Arbor, MI, June 2006.