Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems

Yunhong Zhou, Deeparnab Chakrabarty, Rajan Lukose
HP Laboratories Palo Alto
HPL-2008-9
Feburary 5, 2008*

sponsored search auctions, budget constrained bidding, knapsack problems

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. 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 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.

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. 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 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 auctions generated an estimated $15 billion in revenue globally in 2006, and the global online advertising market is expected to reach $81 billion in 2011. [1] 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, large companies spend billions of dollars each year in marketing with an increasingly large portion of that dedicated to search 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 [15, 29, 21] 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

*Work was mostly done while the author was an intern at HP Labs.
[1] source: Piper Jaffray & Co.

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 [23] 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 and very easy to implement. As an example, 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 $\frac{V}{\Psi(z(t))}$,

where V is the expected value-per-click of the keyword, and $\Psi(z)$ 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 $\{0, 1, \cdots, N\}$ interested in a *single keyword*. Bidder 0 is the default advertiser and his expected value-per-click for this keyword is V. [2] Bidder 0 has a budget of B over time period T (e.g., if T is 24 hours, B is the daily budget), and he wants to maximize his profit over the time period T. Here the budget constraint is a *hard* constraint, in the sense that once exhausted, the budget can not be refilled; budget remaining at the end of period T is forfeited; the bidder leaves the auction once his budget is exhausted.

Bidders bid on the keyword, and are allowed to change their bids at any moment of time. 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, $b_1 \geq b_2 \geq \ldots \geq b_S$ 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 (CTR), denoted $\alpha(s)$, which is defined as the total number of clicks on an ad divided by the total number of impressions (displays). Usually $\alpha(s)$ decreases from upper to lower position slots. [3] Each time his ad in slot s is clicked, bidder 0 is charged a *cost* of b_{s+1}, obtains an expected *revenue* V, and *profit* $V - b_{s+1}$, where b_{s+1} is the bid of the advertiser in the $(s+1)$-th slot or $b_{\min}$ if $s = S$.

The total timespan is discretized into T periods 1, 2, $\cdots$, T such that, within a single time period t, no bidder changes his bid. Moreover, suppose Bidder 0 can make his bid in time period t after seeing all other bidders' bids. [4] Bidder 0 needs to decide, how much to bid at each time period t, to maximize its profit (or revenue) while keeping its total cost within its budget.

1.2 Keyword Bidding 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 [8].) 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 $v(t) = (V - b(t))X(t)\alpha$, where $X(t)$ is the expected number of queries at time period t, and $X(t)\alpha$ is the expected number of clicks at t. Thus, the omniscient bidder should choose time periods $S \subset T$ to maximize $v(S) = \sum_{t \in S} v(t)$ 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 $(v(t), w(t))$ 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 $v(t) = VX(t)\alpha$ for each t. However, for the bidding optimization problem, items arrive in an online fashion. At each time period t, bidder 0 has to decide 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 past decisions. Thus keyword bidding 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 5.

The knapsack problem is a classic problem in operations research and theoretical computer science. For

[2] 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.

[3] Both the expected value-per-click V and the CTR $\alpha(s)$ may vary over different advertisers and different ads. Here we focus on the default advertiser (bidder 0) and assume that both V and $\alpha(s)$ are fixed over time T.

[4] Here we assume that bidder 0 can see other bidders' bids at any time. It had been true until February 2007 through Yahoo!/Overture View Bids tool. At present, for all search advertising platforms, you can only get estimated costs for each position. However, you can always find out the cost for each position by probing the system with different bidding prices and check the corresponding positions you got. The amount of money the system charged you also reveals the cost for your current position.

online knapsack problems, Marchetti-Spaccamela and Vercellis [23] 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. [5] Fortunately, in our setting, we can make two reasonable assumptions on the knapsack items, which allow us to develop interesting online algorithms. We state the assumptions below and justify them in Section 4:

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., $L \leq \frac{v(t)}{w(t)} \leq U, \quad \forall t$.

1.3 Our Results

In Section 3, we design both determinstic and randomized algorithms for the online knapsack problem with two assumptions given above. Both our algorithms have competitive ratio $\ln(U/L) + 1$, while the deterministic algorithm 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.

In Section 4, we show that bidding optimization for single-slot auctions corresponds to the online knapsack problem. We translate the online knapsack algorithm into bidding strategies for the single-slot auction, for both profit and revenue maximization. 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 5, we give a $(\ln(U/L) + 2)$-competitive online algorithm for the multiple-choice knapsack problem (MCKP), a classic generalization of the knapsack problem. We show that the general bidding optimization problem (with multiple slots per keyword) corresponds to the Online-MCKP. We translate the algorithm for Online-MCKP to bidding strategies for the multiple-slot case, and obtain both profit-maximizing 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.

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 [15, 4] to show that the generalized second-price scheme is not truthful.

[5]Consider 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.

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, and it performs much better empirically while maintaining the same theoretical bounds. Our experimental work (restricted by limited evaluation data), reported in Section 6, suggests that parameter tuning also 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. RELATED WORK

Given the numerous research literature in sponsored search auctions, knapsack problems, and online algorithms, we try to discuss these mostly related to our work.

Bidding Optimization in Keyword Auctions. Over the past few years, keyword auctions have attracted a lot of attention both from the auctioneer's perspective ([15, 29, 4, 21]) and the advertiser's perspective ([14, 5, 28, 30]). For auctioneer revenue maximization with budget-constrained bidders, there are various work with various complexities to model keywords, slots, and clicks ([9, 2, 24, 10, 3]). Among them, the techniques used by Mehta etc al. [24] are perhaps most similar 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.

For bidding optimization for the advertiser, Kitts and LeBlanc [18] describe various bidding heuristics. Borgs et al. [7] propose a bidding strategy which over time equalizes the ROI over all keywords. Rusmevichientong and Williamson [27] discuss about how to learn the CTRs for various keywords over time and select keywords accordingly. Most recently, Feldman et al. [16] studied variants of the bidding optimization problem where the objective is to maximize the number of clicks, with possibly complicated interactions among many keywords, and Carey et al. [13] analyzed properties of greedy bidding strategies. None of the previous work has modeled the bidding problem as online knapsack problems, while we design simple threshold based algorithms for online knapsack problems and translate them back to solve the bidding problem.

Online Knapsack Problems. Marchetti-Spaccamela and Vercellis [23] were the first to study the problem and showed that in the general case, there exists no online algorithm achieving any non-trivial competitive ratio. Many special cases of the problem have been studied, including the stochastic online knapsack problem [22, 26, 20], the removable online knapsack problem [17] and the online partially fractional knapsack problems [25], none of them seem to imply our assumptions. A special case of the online knapsack problem where all items have unit cost is the online multiple secretary problem. Kleinberg [19] gave an online algorithm with competitive ration $1-O(1/\sqrt{k})$ for the online k-secretary problem where k is the number of secretaries to select.

Online Call Routing and General Packing Problems. Awerbuch et al. [6] studied the online call routing which generalizes the online classical knapsack problem. More recently, in a series of works Buchbinder and Naor [11, 12] designed online algorithms for fractional versions of general packing problems and derived the results of Awerbuch et al. in their framework. One can possibly derive $O(\ln(U/L))$-competitive online algorithm for the classical knapsack problem from their algorithms.

Our (optimal) competitive ratio of $\ln(U/L)+1$ can be thought of pinning down the exact constant in their results for the knapsack problem via simpler algorithms. 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.

3. THE ONLINE KNAPSACK PROBLEM

The input sequence consists of a knapsack of capacity B and a stream of T items having values and weights $(v(t), w(t))$. We call the value-to-weight ratio $(\frac{v(t)}{w(t)})$ of any item its *efficiency*. The goal is to choose these items in an *online* fashion, that is make a decision as the items come and not revoke them, so as to maximize the total value. 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 $\mathrm{OPT}(\sigma) \le \gamma \cdot \mathcal{A}(\sigma)$, 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 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 $L \le x \le U$, and $f(x) = c/L$ for $0 \le x \le L$, where $c = \frac{1}{1+\ln(U/L)}$ Note that $\int_0^U f(x)dx = 1$, and this is a valid density function.

Algorithm Online-KP-Randomized
Pick a threshold T from the distribution $\mathcal{D}$.
At time t, pick element t iff

$$\frac{v(t)}{w(t)} \ge T \qquad \text{and budget remaining } \ge w(t).$$

Theorem 3.1. Online-KP-Randomized *has competitive ratio* $\ln(U/L)+1$.

Proof. We may assume that the optimum fills the knapsack. Given σ, for $x \in [0, U]$, let $\rho(x)$ denote the fraction of knapsack filled by the optimum algorithm with items whose efficiency ratio is more than x. Note that ρ is a continuous decreasing function with $\rho(0) = \rho(L) = 1$. Assume $\rho(U) = 0$. Since the fraction of knapsack containing elements of efficiency ratio exactly x is $\rho(x) - \rho(x+dx) = -d\rho(x)$, we see

$$\mathrm{OPT}(\sigma) = -\int_0^{\mathrm{U}} \mathrm{x}d\rho(\mathrm{x}) \cdot \mathrm{B} = \int_0^{\mathrm{U}} \rho(\mathrm{x})\mathrm{dx} \cdot \mathrm{B} \qquad (1)$$

where the second equality follows from simple calculus.

Now note that if the random threshold chosen is $T \ge L$, then the algorithm would have a profit of at least $T\rho(T) \cdot B$. Also if $0 \le T \le L$, then the algorithm would have a profit of at least LB.

Thus the expected profit of the algorithm

$$\begin{aligned}
\mathbf{E}[\mathcal{A}(\sigma)] &\ge \int_0^L LBf(x)dx + \int_L^U x\rho(x)Bf(x)dx \\
&= LB\int_0^L \frac{c}{L}dx + B\int_L^U x\rho(x)\frac{c}{x}dx \\
&= cB\int_0^L \rho(x)dx + cB\int_L^U \rho(x)dx \\
&= c \cdot \mathrm{OPT}
\end{aligned}$$

where the last but one equality uses $\rho(x) = 1$ for $x \in [0, L]$. The proof follows by observing $c = \frac{1}{\ln(U/L)+1}$. □

Remark: Note that the algorithm works well only in expectation and against oblivious adversaries. If the threshold choice is known, then an adversary can produce an arbitrarily bad input sequence.

Deterministic Algorithm: Now we state the deterministic algorithm for the online knapsack problem which works against all adversaries 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

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

Observe that for $z \in [0, c]$ where $c \equiv \frac{1}{1+\ln(U/L)}$, $\Psi(z) \le L$, 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.

Remark: The choice of the function Ψ might seem mysterious to readers. Actually, the function was obtained in a systematic fashion via a limiting series of more discrete functions. This method of obtaining the continuous version of a discrete algorithm was also done in [24].

Theorem 3.2. *For any input sequence σ, if $\mathcal{A}(\sigma)$ is the profit obtained by* Online-KP-Threshold *has a* *competitive ratio of and* $\mathrm{OPT}(\sigma)$ *is the maximum profit that can be attained, then*

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

In other words, the above algorithm 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

$$\frac{\mathrm{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{P + \Psi(Z)(B - W)}{P + v(S \setminus S^*)}. \tag{2}$$

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

$$P \geq \sum_{j \in S \cap S^*} \Psi(z_j) w_j =: P_1, \tag{3}$$

$$v(S \setminus S^*) \geq \sum_{j \in S \setminus S^*} \Psi(z_j) w_j =: P_2. \tag{4}$$

Since $\mathrm{OPT}(\sigma) \geq \mathcal{A}(\sigma)$, inequality (2) implies

$$\frac{\mathrm{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{P + \Psi(Z)(B - W)}{P + v(S \setminus S^*)} \leq \frac{P_1 + \Psi(Z)(B - W)}{P_1 + v(S \setminus S^*)} \tag{5}$$

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

$$\frac{\mathrm{OPT}(\sigma)}{\mathcal{A}(\sigma)} \leq \frac{\Psi(Z)}{\sum_{j \in S} \Psi(z_j) \Delta z_j} \tag{6}$$

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

Now 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,

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

and along with inequality (6) 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.

3.1 A Matching Lower Bound

Theorem 3.3. *The competitive ratio of any (possibly randomized) online algorithm for the online knapsack problem is at least* $\ln(U/L) + 1$.

Proof. Yao's minimax principle says for any input distribution $\mathcal{D}$ and any γ-competitive randomized algorithm $\mathcal{A}$,

$$\frac{1}{\gamma} \leq \min_{\sigma} \frac{\mathbf{E}[\mathcal{A}(\sigma)]}{\mathrm{OPT}(\sigma)} \leq \max_{\text{deterministic } A} \mathbf{E}_{\sigma \leftarrow D} \left[\frac{A(\sigma)}{\mathrm{OPT}(\sigma)} \right]$$

To prove the theorem we specify a distribution $\mathcal{D}$ such that

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

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 (we specify p_j's later).

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

$$\begin{aligned}
\mathbf{E}_{\sigma \leftarrow D} \left[\frac{A(\sigma)}{\mathrm{OPT}(\sigma)} \right] &= \sum_{i=0}^{k} \frac{p_i \sum_{j=0}^{i} (1 + \eta)^j f_j}{(1 + \eta)^i} \\
&= \sum_{j=0}^{k} f_j \sum_{i=j}^{k} p_i (1 + \eta)^{j-i}
\end{aligned}$$

Next we specify the p_j's:

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

Notice that $\sum_j p_j = 1$, so the distribution is defined appropriately. In addition, you can verify that

$$\sum_{i=j}^{k} p_i (1 + \eta)^{j-i} = \frac{1 + \eta}{(k + 1)\eta + 1}, \ \forall\ j.$$

Thus we get

$$\mathbf{E}_{\sigma \leftarrow D} \left[\frac{A(\sigma)}{\mathrm{OPT}(\sigma)} \right] = \frac{1 + \eta}{(k + 1)\eta + 1} \sum_{j=0}^{k} f_j \leq \frac{1 + \eta}{k\eta + 1}$$

where the last inequality uses the fact that $\sum_{i=0}^{k} f_i \leq 1$ as the algorithm can not over-fill the knapsack. By setting $\eta \to 0$ and using the fact that $k\eta \to \ln(U/L)$, then Eq.(7) is proved. This completes the proof. $\square$

4. SINGLE-SLOT AUCTIONS

In this section, we translate the algorithms in Section 3 for Online-KP to bidding strategies for single-slot keyword auctions. The bidding optimization objective can be either profit maximization or revenue maximization. Before presenting the algorithms, we first explain why the assumptions made in Section 1.2 are justified.

For profit maximization, recall that the unique item at time period t has a weight $w(t)$ and a profit $v(t)$ where

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

For revenue maximization, $v(t) = VX(t)\alpha$. 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 consider U first. In the case of profit maximization, since $\frac{v(t)}{w(t)} = \frac{V}{b(t)} - 1$, it suffices to set $U \equiv \frac{V}{\mathrm{b_{min}}} - 1$. In the case of revenue maximization, we have $v(t)/w(t) = V/b(t)$, and it suffices to set $U \equiv \frac{V}{\mathrm{b_{min}}}$.

To get a lower bound L for profit-maximization, 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. For 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 the whole budget. Therefore, we only need to consider items with efficiency at least 1, i.e., set $L = 1$.

4.1 Bidding Strategies for Single-Slot Auctions

We now construct bidding strategies for two objective: one to maximize profit and the other to maximize revenue. The difference in the two are in the parameter settings. 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.

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

Note that both the 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 two theorems:

THEOREM 4.1. *For any $\epsilon > 0$,*

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

The proof of Theorem 4.1 follows from Theorem 3.2 and the fact that all items with efficiency $\leq \epsilon$ has total value at most ϵB. Theorem 4.1 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 6.2.

THEOREM 4.2. *Assuming that* OPT *does not overbid at time t where $b(t) > V$, then*

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

The proof of Theorem 4.2 follows from Theorem 3.2 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.

5. ONLINE-MCKP AND MULTIPLE-SLOT AUCTIONS

In this section we first generalize the online algorithm for the knapsack problem to the multiple-choice knapsack problem (MCKP), then translate the algorithm for Online-MCKP to bidding strategies for multiple-slot auctions.

5.1 The Online MCKP

The Online-MCKP is a generalization of the Online-KP. At each time period, *at most* one item is to be chosen from a item set N_t. The goal again is to maximize the total value of items chosen.

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,
> $$E_t \equiv \left\{ s \in B_t \mid \frac{v_s(t)}{w_s(t)} \geq \Psi(z(t)) \right\},$$
> pick element $s \in E_t$ with maximum $v_s(t)$

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

THEOREM 5.1. ONLINE-MCKP-THRESHOLD *has a competitive ratio of* $(\ln(U/L) + 2)$.

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

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

Note that the claim immediately implies the theorem. As in the proof of Theorem 3.2, let S and S^* be the set of items picked by the algorithm and the optimum, respectively. Let $P = v(S \cap S^*)$ 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 3.2, 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 a 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 3.2. A similar proof shows that the above claim holds. □

5.2 Bidding Strategy for Multiple-Slot Auctions

For multiple-slot auctions we consider both profit-maximizing and revenue-maximizing cases. At each time period, bidder 0 has to decide which slot's bidder should he outbid. 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 click through rates increase as we move up the slots, bidding higher would only give a higher revenue.

The parameters are as in the single-slot auction case.

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

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 4.1, the above bidding strategy has a performance guarantee, stated as the following theorem:

THEOREM 5.2. *For any* $\epsilon > 0$,

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

As stated above, 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 in Section 4.1, which has the desirable property of obliviousness.

THEOREM 5.3. *The bidding strategy for single-slot auctions also gives the following guarantee for multiple-slot auctions:*

$$\mathrm{OPT_r} \;\leq\; (\ln(\mathrm{V/b_{min}}) + 2) \cdot \mathrm{Revenue}\ .$$

6. EXPERIMENTAL EXPLORATION

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

6.1 Simulation and the Sniping Heuristic

Consider single slot auction with just one other bidder. Figure 1 below shows the simulation of the bidding strategy against the other bidder who bids uniform random in [4, 6]. While the original strategy attains around 45% of that obtained by the omniscient bidder, the modified strategy raises its price at the end and attains around 67% of the optimum. The modified strategy emulates original strategy up to time period 800 and then raises its price. It seems that the original strategy is too conservative: it spends only 64% of the budget and never increases its bids for 80% of the time.

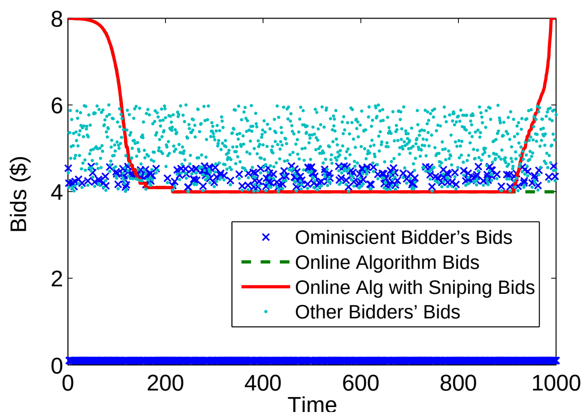

Figure 1: Performance comparison of various bidding strategies in presence of one other bidder who bids a price uniform random in [4, 6].

The reason of course is that the strategy is unaware of the time remaining in the auction. It stops overbidding too early, missing out possible advantageous outbids later on. 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 as follows.

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 $t < \tau \le T$. 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.

> **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 6.1. *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 $t_0 \le T$ exists. Next we claim that $p_2(t)$ is monotone increasing for all $t \ge t_0$. If this is true, then since $p_2(t)$ monotone increasing and $p_1(t)$ monotone decreasing, thus $p_2(t) \ge p_1(t)$ for all $t \ge t_0$.

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

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

Since

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

thus

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

Since $p_1(t)$ is monotone decreasing in t, and $p_2(t)$ is monotone increasing when $t \ge t_0$, 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. □

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 $\rho > \frac{v_s(t)}{w_s(t)}$ & $b_s(t) \le \frac{(1-z(t))B}{\alpha(s)\int_t^T X(\tau)d\tau}$:
> $\quad\rho = \frac{v_s(t)}{w_s(t)}$

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

6.2 Evaluation using Real Bidding Data

Next we report some experimental results on evaluating bidding algorithms for multiple-slot auctions using real bidding data. Due to the lack of publicly available bidding dataset, we *manually* collected bidding prices associated with each position from the Overture webpage [1]. We launched an IE browser visiting Overture view bids website, refreshing the webpage periodically and downloading bidding data from it. Due to the website's anti-frequent-crawling policy, we had to periodically answer Turning tests to keep the crawling process alive. We managed to download data for about two weeks, for one of the most dynamic and expensive keyword "auto insurance." Each time period is about 1 minute which roughly corresponds to the rate at which bids change. There are totally $T = 1842$ distinct time periods in our collected data.

For the experiments, we use $B = 1000$ and three different values $V = 8.0, 10.0, 12.0$. We evaluated both the profit-maximizing and revenue-maximizing strategies with and without sniping. [6] 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. It turns out that tuning the parameter L makes a significant difference. If we choose $L = 0.1$ for profit maximization, we will 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.

Since results are very similar for different parameter values, we summarize them in Table 1. For all the examples we run, sniping improves the bidding performance significantly while exhausting the budget . Table 1 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 Algorithm				
V	OPT	ALG ALG/OPT	budget left	ALG ALG/OPT (sniping)
8	3779	2751 73%	225.5	3541 94%
10	4974	4059 82%	116.1	4607 93%
12	6169	4463 72%	240.8	5842 95%
Revenue-Maximization Bidding Algorithm				
8	4779	3627 76%	195	4505 94%
10	5974	4235 71%	236	5565 93%
12	7169	5081 71%	240	6701 93%

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

7. 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.

We use worst-case competitive analysis, comparing our bidding strategy with the omniscient bidder who know everything 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 bidding algorithms with better performance with the capabilities to learn other agents' bidding strategies or bidding price distributions. Incorporating previous work on stochastic knapsack problems together with average-case analysis (e.g. Lueker [22]) might be an essential ingredient. In a companion paper [31], we try to address these problems.

There is a small gap of 1 in the lower and upper bounds for the competitive ratio of the Online-MCKP. As an open problem, it will be nice to close the gap.

Acknowledgement: We thank the following people for helpful discussions and feedback: Gagan Aggarwal, Terence Kelly, Victor Naroditskiy, Paul O'Brien, David Pennock, Jim Saxe and Bob Tarjan. We also thank an anonymous referee for suggesting randomized threshold algorithms.

[6] We 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.

8. REFERENCES

[1] Overture view bids. `http://uv.bidtool.overture.com/d/search/tools/bidtool`, available until Febuary 2007.

[2] Z. Abrams. Revenue maximization when bidders have budgets. In *Proc. SODA*, pages 1074–1082, 2006.

[3] Z. Abrams, O. Mendelevitch, and J. A. Tomlin. Optimal delivery of sponsored search advertisements subject to budget constraints. In *ACM EC*, 2007.

[4] 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.

[5] K. Asdemir. Dynamics of bidding in search engine auctions: An analytical investigation. In *Proc. 2nd Workshop on Sponsored Search Auctions*, 2006.

[6] B. Awerbuch, Y. Azar, and S. A. Plotkin. Throughput-competitive on-line routing. In *FOCS*, pages 32–40, 1993.

[7] C. Borgs, J. Chayes, O. Etesami, N. Immorlica, K. Jain, and M. Mahdian. Bid optimization in online advertisement auctions. In *Proc. SSA*, 2006.

[8] C. Borgs, J. Chayes, O. Etesami, N. Immorlica, K. Jain, and M. Mahdian. Dynamics of bid optimization in online advertisement auctions. In *Proc. WWW*, pages 531–540, 2007.

[9] C. Borgs, J. T. Chayes, N. Immorlica, M. Mahdian, and A. Saberi. Multi-unit auctions with budget-constrained bidders. In *Proc. ACM EC*, pages 44–51, 2005.

[10] N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proc. ESA*, pages 253–264, 2007.

[11] N. Buchbinder and J. Naor. Online primal-dual algorithms for covering and packing problems. In *Proc. ESA*, 2005.

[12] N. Buchbinder and J. Naor. Improved bounds for online routing and packing via a primal-dual approach. In *Proc. FOCS*, 2006.

[13] M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, and A. R. Karlin. Greedy bidding strategies for keyword auctions. In *ACM EC*, 2007.

[14] B. Edelman and M. Ostrovsky. Strategic bidder behavior in sponsored search auctions. In *Proc. 1st Workshop on Sponsored Search Auctions*, Vancouver, Canada, June 2005.

[15] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 2007.

[16] J. Feldman, S. Muthukrishnan, M. Pal, and C. Stein. Budget optimization in search-based advertising auctions. In *ACM EC*, 2007.

[17] K. Iwama and S. Taketomi. Removable online knapsack problems. In *ICALP*, pages 293–305, 2002.

[18] B. Kitts and B. Leblanc. Optimal bidding on keyword auctions. *Electronic Markets*, 14(3):186–201, 2004.

[19] R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proc. SODA*, pages 630–631, 2005.

[20] A. J. Kleywegt and J. D. Papastavrou. The dynamic and stochastic knapsack problem. *Operations Research*, 46(1):17–35, 1998.

[21] 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.

[22] G. S. Lueker. Average-case analysis of off-line and on-line knapsack problems. *J. of Algorithms*, 29(2):277–305, 1998.

[23] A. Marchetti-Spaccamela and C. Vercellis. Stochastic on-line knapsack problems. *Mathematical Programming*, 68:73–104, 1995.

[24] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. Adwords and generalized on-line matching. In *Proc. FOCS*, pages 264–273, 2005.

[25] J. Noga and V. Sarbua. An online partially fractional knapsack problem. In *Proc. 8th Sym. Parallel Architectures, Algorithms and Networks*, pages 108–112, 2005.

[26] J. D. Papastavrou, S. Rajagopalan, and A. J. Kleywegt. The dynamic and stochastic knapsack problem with deadlines. *Management Science*, 42(12):1706–1718, 1996.

[27] 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.

[28] 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.

[29] H. R. Varian. Position auctions. *International Journal of Industrial Organization*, 2007, to appear.

[30] Y. Zhou and R. Lukose. Vindictive bidding in keyword auctions. In *Proc. 9th Int. Conf. Electronic Commerce*, pages 141–146, Minneapolis, MN, August 2007.

[31] Y. Zhou and V. Naroditsky. Algorithm for online multiple-choice knapsack problems and applications to bidding in keywords auctions.