Semester Project: Bidding Agent for Advertisement Auctions

Luděk Cigler

Abstract

Several bidding strategies for sponsored search auctions have been recently described in the literature. In this work, I show how they perform empirically and against an omniscient strategy. Also, I present a new profit maximizing strategy for multiple keyword ad campaigns which takes into account the budget constraint. For small budgets, it outperforms the other strategies and it remains competitive also for larger budgets.

1 Introduction

Sponsored search is a major source of revenue to search engines, such as Google and Yahoo!. It is also a new opportunity for advertisers. By creating targeted advertisements for specific search keywords, they can focus directly on those potential customers who are looking exactly for what they offer.

However, managing a sponsored search campaign is not an easy task. The advertisers have to choose keywords which are relevant to their offering and write effective ads which would bring more customers. Furthermore, to optimize the cost of the campaign and to make sure their ads get enough attention, they have to set the right *bids* in a keyword auction. If they set their bids too low, the ads might end up not being shown on at all. If the bids are too high, the cost of a click may be higher than the actual revenue. Also, the advertisers usually have limits of how much they can spend during a period of time.

Most of the (published) research in sponsored search has viewed the problem from the perspective of an auctioneer (search engine). Using game-theoretical analysis, people have analysed the various equilibriums of the game: Nash equilibrium ([CDE+08]), *localy envy-free* equilibrium (in which bidders do not want to change their *advertising slots* rather than their bids, [EBO+07]) or *market equilibrium* ([BCI+07]). Optimisation of advertisers' profits has been done mostly by hand, using more-or-less established practises. In a rare publication from this area, Kitts and LeBlanc in [KL04] describe an automated system to optimise sponsored search campaigns. However, to the best of my knowledge, there has been very little research done on comparing the behaviour of various bidding strategies from the perspective of the advertisers.

This work tries to look at the bidding strategies which have been described so far, compare its performance and discuss some of their issues. It is organised as follows: In section 2, I first formally describe strategies which I have implemented and present a new profit maximising strategy. In section 3 I present an environment for evaluating bidding strategies, together with actual experiments. Finally, section 4 concludes.

2 Bidding Strategies

I have focused on the following strategies previously described in literature:

Return-on-Investment (ROI) [BCI+07] On each day t, all bids of an advertiser i are determined by a single parameter $R_i(t) \in (0, 1]$. The parameter $R_i(t)$ is adjusted based on the performance of advertiser i on the previous day. Starting from an arbitrary $R_i(0) \in (0, 1]$ for day $t = 0$, the advertiser i sets

$$R_i(t+1) = \begin{cases} R_i(t) \cdot e^{-\epsilon} & \text{if } i \text{ runs out of money early on day } t \\ \min(R_i(t) \cdot e^{\epsilon}, 1) & \text{otherwise} \end{cases}$$

where $\epsilon > 0$ is a small constant. Finally, she sets the bid $b_{ij}(t)$ for a keyword j to

$$b_{ij}(t) = R_i(t) \cdot u_{ij}$$

Knapsack-ROI [BCI+07] This strategy requires the knowledge of slot prices. Let u_{ij} be the utility of a click on keyword j for advertiser i, p_{js_j} the payment for a click on keyword j in slot s_j. The strategy tries to maintain an invariant that for some constant R_i (the return-on-investment), it holds that $R_i \in (u_{ij}/p_{js_j}, u_{ij}/p_{js_j+1})$ for all keywords j. Note that since we assume that worse slots (with higher index) have lower price, it holds that $u_{ij}/p_{js_j} < u_{ij}/p_{js_j+1}$. The algorithm searches for the minimum R_i such that the budget constraint is not exceeded.

The algorithm monitors what portion of the budget was spend each day. If the advertiser i ran out of money before the end of the day, it looks for a keyword j for which $R_{ij} = u_{ij}/p_{js_j}$ is minimal and where s_j is not the worst slot, and chooses to bid so as to get slot $s_j + 1$ for keyword j on the following day. This decreases spending, but increases R_{ij}. If the advertiser had money left at the end of the day, it finds a keyword j for which u_{ij}/p_{js_j} is maximal and where $s_j > 0$, and chooses to bid for slot $s_j - 1$ on the following day. This should increase spending.

Balanced Best-response [CDE+08] The *Balanced bidding strategy (BB)* is a strategy of advertiser i such that, given other advertisers' bids b_{-i},

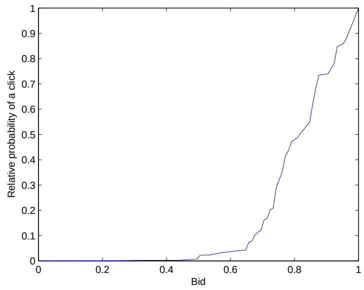
- targets the slot s^*_{ij} for keyword j which maximises his utility, that is

  $$s^*_{ij} := argmax_s\{\theta_s(v_{ij} - p_s(i))\}$$

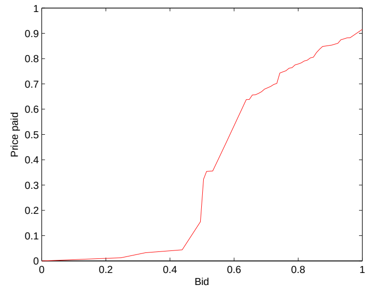
  where θ_s is the click-through rate in slot s, v_{ij} is a utility on click for keyword j and advertiser i and $p_s(i)$ is the price advertiser i needs to pay to get slot s.

- chooses her bid b'_{ij} for the next round to satisfy

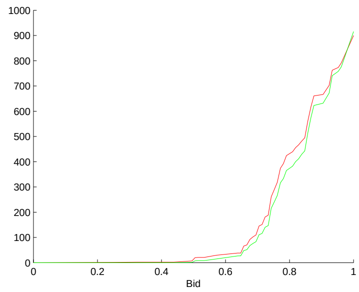
  $$\theta_{s^*_{ij}}(v_{ij} - p_{s^*_{ij}}(i)) = \theta_{s^*_{ij}-1}(v_{ij} - b'_{ij})$$



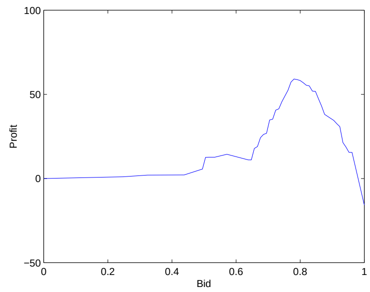
(a) Click probability per bid



(b) Price paid per bid



(c) Revenue (red) and spending (green) per bid, 1000 queries



(d) Profit per bid, 1000 queries

Figure 1: Profit Maximising Strategy Justification

If s^*_{ij} is the top slot, the bid is arbitrarily chosen to be $b'_{ij} := (v_{ij} + p_1(i))/2$.

Online Knapsack [ZCL08] Fix $\epsilon > 0$. Let $\Psi(z) := ((v_{max} \cdot e)/\epsilon)^z(\epsilon/e)$, where v_{max} is the maximum click utility an advertiser has over all of her keywords.

At time t, let $z(t)$ be the fraction of budget spent of a given advertiser, $b_s(t)$ a bid an advertiser must make at time t to win the slot s. Choose slots

$$E_t := \left\{ s | b_s(t) \leq \frac{v_{\max}}{1 + \Psi(z(t))} \right\}$$

Then that advertiser bids $b_s(t)$, where $s := \text{argmax}_{s \in E_t}(v_j - b_s(t))\theta_s$.

2.1 Profit Maximising Strategy

The ROI strategy and the Knapack-ROI strategies described above try to maximise the *revenue* (total value of clicks) an advertiser gets. If the advertiser always spent the whole budget, this would be equal to maximising the profit (total value of clicks minus total spending). However, spending the whole budget is not always desirable to maximise profit.

Suppose that the relationship between a bid for a particular keyword and click probability is as shown on Figure 1a (this graph is taken from one of the simulations, and the click-through probabilities are based on [Bro04]). The price paid for a particular bid is shown on Figure 1b. Then the expected spending for a particular bid corresponds to a product of click probability, price per click and expected number of queries for the keyword, and is shown on Figure 1c. However, if we assume that the utility of a click is constant (does not depend on the slot), the total expected revenue corresponds to a product of click probability, expected number of queries for the keyword and click utility (also shown on Figure 1c for click utility 0.9). Therefore the expected profit per bid is as shown on Figure 1d. We can see that the most profitable bid for an advertiser with utility of a click 0.9 is about 0.8.

If we know the aggregated click-probability-per-bid curve $c_k(b)$ and price-per-bid curve $p_k(b)$ for each keyword k over some period (e.g. one day), we can compute the optimal bid b_k^* for that keyword and day as:

$$b_k^* := \text{argmax}_b (v_k - p_k(b)) \cdot c_k(b)$$

where v_k is the utility of a keyword.

However, such bids can lead to exceeding the budget. In that case, the ads would stop displaying. The problem is that we would receive less clicks for our budget than if we had bid lower in the first place, and those clicks would be more expensive. Therefore it is necessary to decrease our bids so that (in expectation) we spend the budget just at the end of the day - we'll get higher number of clicks, each for lower price.

In case there are more keywords we are bidding on, there are multiple options on which bid to decrease. The bids are descreased by an iterative greedy algorithm: in each iteration, we decrease the bid of a selected keyword by \$0.01. The keyword for which the decrease in expected profit is the lowest is selected. When the total expected spending

$$\sum_{k \in K} p_k(b_k) \cdot c_k(b_k) \cdot h_k$$

(where b_k is the current bid for a keyword k and h_k the expected number of queries for keyword k per day) decreases below the budget limit, the algorithm stops. The computed bids are then used on the following day.

The bids are updated every day based on fuctions p_k and c_k estimated from the auctions of the previous day.

2.2 Optimal Strategy

To compare the performance of various strategies with a theoretical optimal strategy, I use multiple-choice knapsack heuristic described in [FS93]. The heuristic creates a single knapsack item for each (slot, keyword) pair with weight equal to the expected day-long price paid for that slot and keyword $p_{s_{kj}} \cdot q_k \cdot \theta(s_{kj})$; there $p_{s_{kj}}$ is the slot price, q_k number of queries for a keyword k per day and $\theta(s_{kj})$ is the click-through-rate for the slot s_{kj}. The profit from a knapsack item is the expected day-long profit $u_k \cdot q_k \cdot \theta(s_{kj})$ (u_k is the utility of a click for the advertiser).

The knapsack items are sorted descendingly according to the profit/weight ratio, and added to the knapsack greedily as long as they fit into the budget (capacity). In case we try adding a (slot $= s_i$, keyword $= k$) pair for a keyword k which has already another slot s_j in the knapsack, we replace it with the slot s_i provided the slot s_i has higher profit and fits into the knapsack with s_j removed.

3 Experiments

3.1 Simulation Environment

For the purpose of simulations, I have developed a simulation environment to run various strategies. It uses generalised second price rule to clear the auctions. Clicks are generated according to the separability assumption, i.e. that the probability of a click in one slot does not depend on ads placed in other slots. Also, I assume that the click-through probability does not depend on the advertiser. The click-through probabilities are set to values described in [Bro04]. The whole simulation environment is implemented in Python.

3.2 Experimental Scenarios

In the experiments, I used three different keyword sets: with 1 keyword, 20 keywords and 40 keywords. The keyword sets are created using Google AdWords Keyword Tool ([Goo]). For a given phrase, this tool generates a set of suggested keywords K. For each keyword $k \in K$, Google provides approximate number of queries in the last month q_k and a *competition factor* $c_k \in [0, 1]$. The higher the c_k is, the higher the competition is (the number of advertisers bidding for that keyword).

The competition factor c_k is used to determine which advertisers are going to bid for keyword k. For simulation with N advertisers, $N_k = \lceil (N-1) \cdot c_k \rceil + 1$ randomly selected advertisers bid for keyword k.

Bidder valuations for a keyword k are generated randomly from Gaussian distribution with the mean:

$$\mu_k = \left(1 - \frac{1}{\max_{l \in K} q_l}\right)^{q_k} \cdot \mu^*$$

and standard deviation

$$\sigma_k = \frac{\log q_k}{\log(\min_{l \in K} q_l + 1)} \cdot \sigma^*$$

where μ^* and σ^* are parameters of the simulation. Valuations are specified *per click*.

This way, the valuation distribution fulfils the following objective:

- If the number of queries for a keyword is high, it probably means that it is a very general keyword. For such keyword, the average profit is likely to be smaller (the searchers are not targeted very well towards buying an item). On the other hand, for very rare keywords, the searches are probably quite targeted, and it might be possible to get higher profit from them.

- For a keyword with higher number of queries, there is a large room for variations (because there are so many options to write an ad for that keyword). Therefore, the variance should be larger. For a very specific (rare) keyword, the room for deviations in profits is smaller (e.g. the margins on a particular model of digital camera are likely to be very similar).

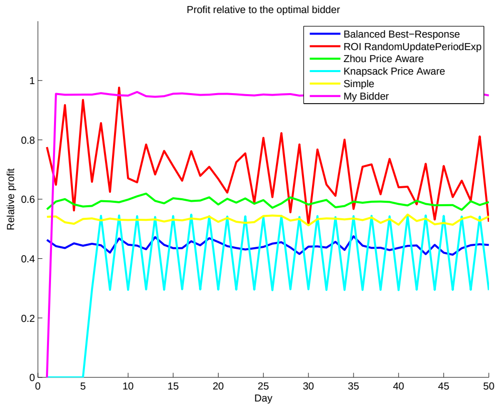
In all experiments, there are in total 30 advertisers and 15 slots: 5 advertisers from each of the described strategies (ROI, Knapsack-ROI, Balanced Best-response, Online Knapsack and Profit Maximising strategy), together with 5 bidders of a *simple strategy*. The simple strategy is to bid just one half of one's keyword utility, as long as the budget is not exceeded. Budgets of all bidders in one simulation are the same.

3.3 Discussion of the Results

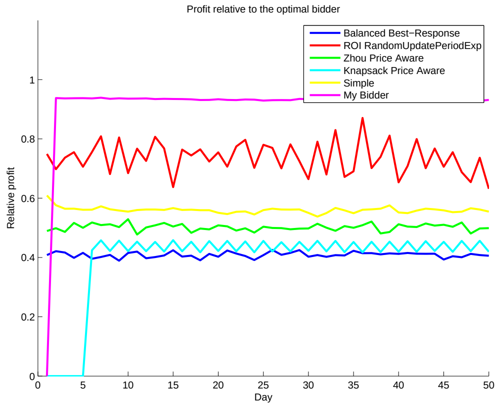
Each experiment has simulated 50 days of search traffic, and has been run 10 times. The graphs on Figure 2, Figure 3 and Figure 4 show average relative profits of different strategies when compared to the profits of an optimal bidder (see section 2.2). The average is taken over all bidders of the same strategy in all runs.

For a single keyword case, I have set bidders' budgets to three different levels: \$2, \$10 and \$100. The profit maximising strategy stays within 90% of the performance of the optimal bidder for lower budgets, and for higher budget, it's performance drops to some 40% of the optimum. The ROI strategy has larger variance in performance between each day, but generally stays among the top performing strategies. The Balanced best-response strategy stays around the same percentage of the optimal strategy for all the budget settings. This may be probably caused by the fact that this strategy does not take the budget into account at all. Relatively stable performance (without much influence of the budget settings) can be observed also at the Knapsack-ROI strategy. All the described strategies gain advantage against the simple strategy as the budget grows, presumably because with larger budget, it becomes more advantageous to bid higher portion of one's true click utility.

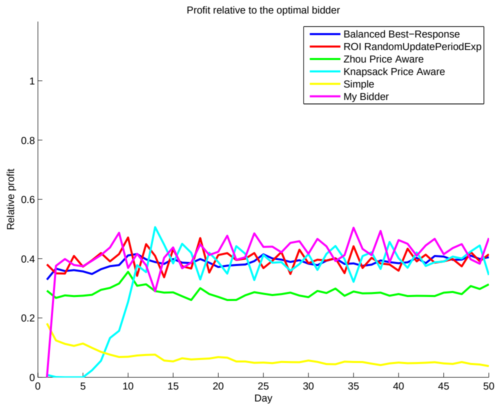
For a 20-keyword case, the budget levels were \$10, \$100 and \$500. Results of the different strategies were in general similar to the 1-keyword case, with



(a) Budget \$2



(b) Budget \$10



(c) Budget \$100

Figure 2: Profit relative to the profit of an optimal strategy, 1 keyword

the exception of Knapsack-ROI strategy, which performed better for the case of low budget. This might be caused by the fact that in 1-keyword case, the strategy has very little opportunities to adjust it's bids to spend the budget precisely (because it can only change the slot of one keyword), while with many keywords there are more options in adjusting the bid vector.

For a 40-keyword case, the budget levels were also \$10, \$100 and \$500. The performance of the various strategies remained similar to the 20-keyword case.

4 Conclusions

In this work, I have looked at various bidding strategies and compared them in several experimental scenarios based on the profits they bring. I compared their performance to an optimal "omniscient" strategy which knows all the bids of all bidders in advance and can choose a different bid for each auction.

I have designed a new profit maximising strategy which takes into account campaigns with multiple keywords and limited budget. The experiments show that this strategy performs within 90% of the optimal strategy for low-buget settings, and has comparable performance to the other best performing strategies (ROI, Balanced Best-response) in settings where advertisers' budgets are higher.

There remain several open issues. Some of the assumptions several strategies used are too strong to be found in real world. Most notably, search engines do not provide the advertisers detailed information about click-through rates in different slots or about prices paid for each slot. Therefore, a realistic strategy would have to be able to learn this information as it bids (Kitts and LeBlanc have tried it in [KL04]). Also, the question is how reliable predictions can be drawn from the past in a market as volatile as auctions.

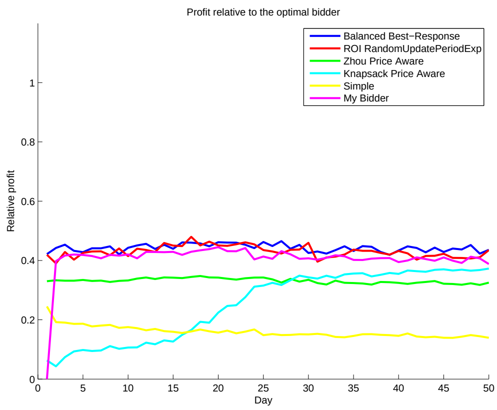
References

[BCI+07] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian. Dynamics of bid optimization in online advertisement auctions. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 531–540, New York, NY, USA, 2007. ACM Press.

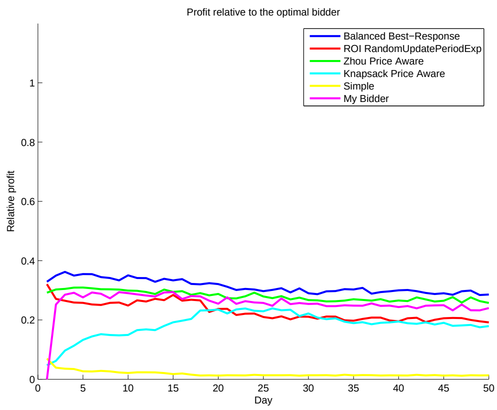
[Bro04] N. Brooks. The Atlas rank report: How search engine rank impacts traffic. *The Atlas Institute*, 2004.

[CDE+08] Matthew Cary, Aparna Das, Benjamin Edelman, Ioannis Giotis, Kurtis Heimerl, Anna R. Karlin, Claire Mathieu, and Michael Schwarz. On best-response bidding in gsp auctions. *National Bureau of Economic Research Working Paper Series*, pages 13788+, February 2008.

(a) Budget $10



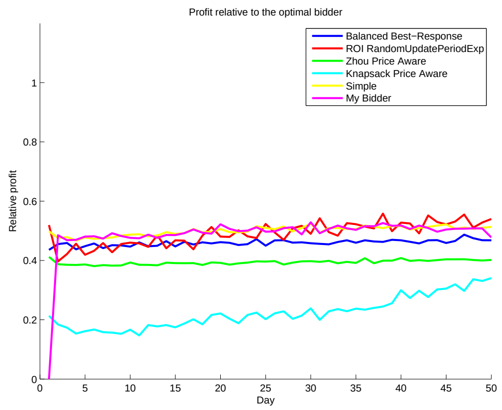
(b) Budget $100



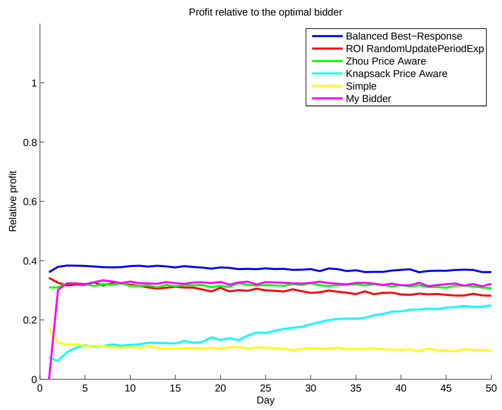
(c) Budget $500

Figure 3: Profit relative to the profit of an optimal strategy, 20 keywords

(a) Budget \$10



(b) Budget \$100



(c) Budget \$500

Figure 4: Profit relative to the profit of an optimal strategy, 40 keywords

[EBO+07] Edelman, Benjamin, Ostrovsky, Michael, Schwarz, and Michael. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American Economic Review*, 97(1):242–259, March 2007.

[FS93] Thomas A. Funkhouser and Carlo H. Séquin. Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 247–254, New York, NY, USA, 1993. ACM.

[Goo] Inc. Google. Google AdWords Keyword Selector. `https://adwords.google.com/select/KeywordToolExternal`.

[KL04] Brendan Kitts and Benjamin Leblanc. Optimal bidding on keyword auctions. *Electronic Markets*, 14(3):186–201, September 2004.

[ZCL08] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1243–1244, New York, NY, USA, 2008. ACM.