# Adwords Auctions with Decreasing Valuation Bids

Aranyak Mehta [*] Gagan Goel [†]

**Abstract**

The choice of a bidding language is crucial in auction design in order to correctly capture bidder utilities. We propose a new bidding model for the Adwords auctions of search engine advertisement – *decreasing valuation bids*. This provides a richer language than the current model for advertisers to convey their preferences. Besides providing more expressivity, our bidding model has two additional advantages: It is an *add-on* to the standard model, and retains its simplicity of expression. Furthermore, it allows efficient algorithms – we show that the greedy (highest bid) algorithm retains its factor of 1/2 from the standard bidding model, and also provide an optimal allocation algorithm with a factor of 1-1/e (as is case in the standard bidding model).

We also show how these bidding languages achieve a good trade-off between expressivity and complexity – we demonstrate a slight generalization of these models for which the greedy allocation algorithm has an arbitrarily bad competitive ratio.

[*] Google, Inc., Mountain View, CA
[†] Georgia Institute of Technology, Atlanta, GA

# 1 Introduction

One of the most important design parameters in auction design is the choice of a bidding language. This is the *interface* provided to the bidders by the seller, which allows them to express their preferences to the seller. There is always a trade-off in this choice. The more complex a bidding language is, the better can it capture bidder preferences, and indirectly, the better it is for the seller. But at the same time, it is essential to have a simple bidding language, so that the bidders will be able to translate their innate preferences into the language in the first place. Simplicity of expression is not the only reason preventing us from choosing highly complex bidding languages. A second reason is *computational*: Even if the bidders express their preferences in a complex bidding language, it may be impossible for the seller to process such complex preferences and decide on an optimal (or even a good) outcome efficiently. Such a trade-off becomes apparent in the design of complex auctions, and has been studied in detail, for example, in the case of combinatorial auctions [4].

In this paper we are interested in the study of the Adwords auctions of search engines such as Google, Yahoo! and MSN with respect to the issue of bidding languages. These auctions, which account for a large portion of the extremely high revenues of these companies, sell keyword search queries to interested advertisers. These are online auctions, in which the bidders (advertisers) express their preferences to the search engines in advance, and as users enter keyword queries, the search engine decides whose ad to display with the search results.

The bidding languages currently provided by the search engines are of the following form: A bidder $i$ can bid, for each keyword $q$ he is interested in, a monetary bid $b_{iq}$, expressing the value he gets if his ad is displayed with search results for queries of type $q$. Together with the bids, the bidder is also allowed to report a *daily global budget* $B_i$, which means that over all the keywords, he is not willing to spend more than $B_i$.

While this bidding model does capture the essential features of advertiser preferences, namely, the individual bids and a global budget, it fails to express more complex constraints that the advertisers may have. For example, the advertiser may not want to be in a situation in which he wins many ad slots, but all of them for queries of the same single keyword. He would prefer to have diversity, in the sense of winning a reasonable amount of ad slots for several different keywords of his choice. There are two reasons to expect real advertisers to have such preferences: firstly, advertisers may wish to make their presence felt in several different sub-markets at the same time, and sell different products at comparable rates. Secondly, there are situations in which advertisers have decreasing marginal utility from subsequent advertisements, e.g., once the ad reaches a certain fraction of the target audience.

It is important to find expressive, yet simple and practically implementable bidding languages which would provide bidders with more control to express such preferences. Note that it is not even possible to simulate such preferences in the current bidding model, say by splitting into several different accounts. We stress the following four **properties of our model:** *Firstly*, our bidding model is an *add-on* to the current model, and hence can be gradually introduced on top of the current model. Bidders may choose to continue bidding as they did in the current model if they prefer. *Secondly*, as our results show, there may be no need to change the allocation algorithms used by the search engine, even upon introducing the new bidding model. *Thirdly*, the better expressivity will, in our opinion, allow the bidders to bid with more control and less risk, and therefore more aggressively, indirectly improve the revenue of the search engine. *Finally*, we believe that this model may lead to less fluctuations in the bids, as opposed to the current model in which bidders may dynamically change their bids as they win certain number of queries.

We note that the importance of expressiveness to achieve efficiency and increased revenue has been studied earlier in the context of Ad auctions [5]. A notion of *spending constraints* was introduced in [6],[1] in the context of market equilibrium. There the utility of a bidder for the next item of a good depends on how much *money* he had already spent on that good. In the first price auction setting, spending constraints would translate to the constraints used in our setting.

## 2 A new bidding language: Decreasing valuation bids

In this paper we propose a new bidding language for the Adwords auctions, designed to express the type of preferences outlined above. We state the model in full generality, but also specify an important special case which is practical, simple to use (e.g. via a simple GUI), and has low communication overhead.

In the *decreasing valuations bid* model, bidder $i$ bids the following:

- A global daily budget $B_i$.
- For each keyword $w$ he is interested in, a *decreasing function* $f_{i,w} : Z^+ \to \mathbf{R}$, which is to be interpreted as follows: If bidder $i$ has already been allocated $x$ number of queries of keyword $w$, then his bid for the next query of keyword $w$ is $f_{iw}(x)$.

Note that the current model only allows constant functions $f_{iw}(x) = b_{iw}, \forall x$. A simple and practical special case of our model is one which allows only functions of the form

$$f_{iw}(x) = \begin{cases} b_{iw} & \text{if } x \leq t_{iw} \\ 0 & \text{otherwise} \end{cases}$$

This special case means that bidder $i$ values each of the first $t_{iw}$ queries of keyword $w$ at $b_{iw}$ each, but does not want more than $t_{iw}$ of $w$'s . We shall call this special case the case of $q$-budgets.

## 3 Results in the new models

Better expressivity is clearly better for the bidder (as long as the language remains simple enough to understand). So with the introduction of this new models of bidding languages, the question which arises naturally is: How does this affect the search engine's profits?

Intuitively, it is clear that the bidders will now be able to bid with more control and therefore face less risk, and will bid more aggressively. This is clearly better for the search engine, in terms of the optimal profit (OPT) derivable from the bidders. But what if the bidding language introduces computationally difficult problems for the search engine? Then it will not be able to efficiently extract a good portion of the OPT as profit. We show that our models do not introduce such computational difficulties, by describing optimal algorithms in both these settings, whose competitive ratio (in an online competitive analysis model) is $1 - 1/e$, as good as that of the optimal algorithm [3] in the standard model.

We also show that our bidding language is at the correct trade-off point between expressivity, simplicity and computational efficiency. Simple generalizations of our bidding model (by adding more expressivity) result in computational problems for which no algorithms can perform better than a factor of 1/2, and for which the natural Greedy algorithm has an arbitrarily bad factor.

## 3.1 Our Techniques

The algorithm we analyze in the new model is precisely the algorithm from [3] for the standard bidding model. For each arriving query, this algorithm (which we will call MSVV in the sequel) determines the *effective bid* of each bidder as his bid for the query scaled by a function $\psi$ of the fraction of budget that the bidder has spent so far (the function is $\psi(x) = 1 - e^{-(1-x)}$). Then the algorithm awards the query to the bidder with maximum effective bid. It is shown in [3] that this algorithm has a competitive ratio of $1 - 1/e$ in the standard bidding model, and that this is optimal (even over randomized algorithms). We show that the same algorithm has the same factor even in our generalized bidding models. Clearly it is optimal since our models are more general than the standard model.

Our proof technique follows the proofs in [3]. In that proof, the main idea was to show that for each query $q$, the algorithm gets some effective amount of money (which is the real money scaled by some factor depending on $\psi$) which is comparable to an effective amount that OPT gains for $q$. This kind of query-by-query analysis is not possible here, since the bid of a bidder $i$ for a query $q$ itself depends on how many other queries of that type have already been allocated to him. Thus the bid depends on the *context* in which $q$ arrives with respect to the previous choices of the algorithm. We take care of this by a careful charging argument: we demonstrate the existence of a map between the queries that OPT assigns and the queries that ALG assigns (not necessarily to the same bidder). This helps us show sufficient profit for ALG. The analysis in [3] can be thought of as the special case when this map is the identity map.

## 3.2 Intuition

At first thought it seems very surprising that this algorithm would perform well in the presence of extra constraints. After all it seems to be ignoring the information provided by, say, the $q$-budgets for different $q$, while OPT may be using this information to its benefit. The reason why the algorithm works is that these particular extra constraints that we add turn out to be fortuitously geared to help the algorithm. The following example illustrates this in the context of the Greedy algorithm (recall that this has factor 1/2 in the standard model):

Consider the example shown in [2, 3] on which the greedy has a competitive ratio of 1/2 (for the standard bidding model). There are two bidders $b_1$ and $b_2$. Bidder $b_1$ is interested in keywords $w_1$ and $w_2$, whereas bidder $b_2$ is only interested in keyword $w_2$. Both have a budget of $N$, and a bid of 1. Now consider the sequence in which $N$ queries of keyword $w_2$ come first and then $N$ queries of keyword $w_1$ come next. $OPT$ will allocate all the $w_2$'s to bidder $b_2$ and all the $w_1$'s to bidder $b_1$, thus earning $2N$ units of money. Whereas greedy could assign all the $w_2$'s to bidder $b_1$, and after that when the $w_1$'s come in bidder $b_1$ cannot buy them, as he has exhausted his budget. So all the $w_1$'s will go wasted, thus greedy will earn only $N$ units of money, or about half of $OPT$.

The reason why greedy is factor 1/2 is that we allocate too many $w_2$'s to bidder $b_1$. Now consider the following two cases in the $q$-budget version.

***Case 1:*** $b_1$ has a $w_2 - budget$ constraint. Now, clearly there is a limit to which greedy could go bad as it cannot allocate *too many* $w_2$'s to bidder $b_1$. So adding a $w_2 - budget$ to the standard model cannot make greedy worse.

***Case 2:*** $b_1$ has a $w_1 - budget$ constraint. This case seems more dangerous, because greedy could do badly by allocating too many $w_2$'s to $b_1$. But this is not the case, since greedy's allocation becomes bad only when $OPT$ allocates too many $w_1$'s to $b_1$. But a $w_1 - budget$ would not allow $OPT$ to gain too much in this case.

Thus in one case, the $q$-budget prevents Greedy from taking a bad step, and in the other case it makes sure that what Greedy missed was not useful anyway. In fact we will show in section 4 that greedy still retains the factor 1/2 in these models. The above ideas, though applied to Greedy, also hold for MSVV.

# 4 The Decreasing Valuations Bidding Language

## 4.1 A Warm-up: Analysis of Greedy

Recall that Greedy was factor 1/2 in the standard model. Does the factor change in the model with decreasing bids? We show that this is not the case – Greedy retains its factor of 1/2.

**Theorem 1** *The competitive ratio of Greedy algorithm in the decreasing valuation bid model is 1/2.*

**Proof :** Let $S_i$ be the set of queries assigned by OPT to bidder $i$. For any set of queries $Q$, let $greedy(Q)$, $OPT(Q)$ denote the money earned by the queries in $Q$ in Greedy and OPT respectively. We will use $greedy(i)$, $OPT(i)$ to denote the money spent by bidder $i$ in *greedy* and OPT respectively. We will show that

$$\forall\, i: \quad greedy(i) + greedy(S_i) \geq OPT(i)$$

Summing over all $i$, we get:

$$\sum_i (greedy(i) + greedy(S_i)) \geq OPT$$

Now any query $q$ can be counted at most twice in the summation: $\sum_i (greedy(i) + greedy(S_i))$. Hence, $2 * greedy \geq OPT$, as required.

Now it remains to prove the equation above. If $greedy(i) \geq OPT(i)$, then we're done. So assume that $greedy(i) < OPT(i)$. This means that $greedy(i) < B_i$, i.e., bidder $i$ has remaining budget even at the end of Greedy.

Let $S_i^w$ be the restriction of $S_i$ to keyword $w$. We only need to consider keywords $w$ s.t. the number of $w$-type queries allocated by greedy to bidder $i$ (call it $num_g$) is less than that allocated by OPT to $i$ (call it $num_o$). Then, for at least $num_o - num_g$ queries from $S_i^w$, greedy assigns there queries to some bidder other than $i$, who bids more for it when it arrives. Also, since the bid curve is non-increasing, for each of these queries, bidder $i$ bids at least as much during greedy as OPT makes for it. Hence greedy obtains at least as much money for each of these queries as OPT makes for it. Therefore, the revenue of greedy from these queries collectively is at least $(OPT(i, w) - greedy(i, w))$ (where $OPT(i, w), greedy(i, w)$ is the total money spent by bidder $i$ on all queries of keyword-type $w$ in OPT and greedy respectively). So $greedy(S_i^w) \geq (OPT(i, w) - greedy(i, w))$, and, summing over all keywords $w$, we get $greedy(S_i) \geq (OPT(i) - greedy(i))$, which is what we wanted to prove. Note how we have crucially used the fact that the bid-curves are non-increasing.

□

## 4.2 Analysis of the MSVV Algorithm

In this section we will analyze the performance of the MSVV algorithm in the decreasing bids model. Let us recall the algorithm in the standard model (without decreasing bids). For clarification, we will use $w$

to name a keyword, and $q$ to name a query – a query $q$ can be of type $w$.

**The Algorithm:** For the next query $q$ (of type $w$) compute the *effective bid* of bidder $i$ as: $b_{iw}\psi(y)$ where $y$ is the fraction of budget spent by $i$, and $\psi(y) = 1 - e^{-(1-y)}$. Award $q$ to the bidder with the highest effective bid.

In the decreasing bids model, the effective bid becomes:

$$f_{iw}(x)\psi(\text{fraction of budget spent by } i)$$

where $x$ is the number of queries of keyword $w$ already allocated to $i$.

We will prove the following theorem:

**Theorem 2** *MSVV achieves a factor of* $1 - 1/e$.

We will follow the proof structure as in [3]. The crucial difference in the proof is in a careful charging via a well-chosen map between queries. We start with some preliminary notation.

Start by picking a large integer parameter $k$. Define the *type* of a bidder according to the fraction of budget spent by that bidder at the end of the algorithm BALANCE: say that the bidder is of type $j$ if the fraction of his budget spent at the end of the algorithm lies in the range $((j-1)/k, j/k]$. Slab $i$ is the portion of money $[(i-1)/k, i/k]$ of all the bidders.

Also, let $\alpha_j$ denote the number of bidders of type $j$. Let $\beta_i$ denote the total money spent by the bidders from slab $i$ in the run of the algorithm. It is easy to see that $\beta_1 = N/k$, and

$$\forall\ 2 \le i \le k, \qquad \beta_i = N/k - (\alpha_1 + \ldots + \alpha_{i-1})/k \tag{1}$$

Let $ALG(q)$ ($OPT(q)$) denote the revenue earned by the algorithm (OPT) for query $q$. Say that a query $q$ is of *type* $i$ if OPT assigns it to a bidder of type $i$, and say that $q$ lies in *slab* $i$ if the algorithm pays for it from slab $i$.

This concludes the notation from [3]. Fix a keyword $w$, and let $Q_w$ be the set of all queries of keyword $w$, and let $Q_w^{OPT}$ be the set of queries of keyword $w$ assigned by OPT to all the bidders of type *strictly less* than $k$ (these are the bidders who haven't spent all their money). We will drop the subscript $w$ when it is clear by context. We will use subscript $i$ to denote the restriction of any variable to bidder $i$.

Let $q_{ij}$ denote the $j^{th}$ query of keyword $w$ allocated to the bidder $i$. Since $f_{i,w}$ is a decreasing function, therefore $f_{i,w}(q_{ij}) \le f_{i,w}(q_{i(j-1)})$. Let $q_{ij}^O$, $q_{ij}^A$ denote the allocation by OPT and ALG respectively. The inequality above holds for both these allocations.

**Lemma 3** *For each keyword $w$, there exists a injective map $\sigma : Q_w^{OPT} \rightarrow Q_w$ s.t. $\forall q \in Q_w^{OPT}$: $OPT(q)\psi(type(q)) \le ALG(\sigma(q))\psi(slab(\sigma(q)))$*

**Proof :** To prove the lemma, we will construct the mapping for every query in $Q_w^{OPT}$ such that the lemma holds (we will drop the $w$ subscript henceforth).We do this in two phases.

**Phase 1:** For each bidder $i$, define
$x_i = min(|Q_i^{OPT}|, |Q_i^{ALG}|)$. We define a mapping for the first $x_i$ queries in the OPT allocation:

$$\forall\ t \in [1, x_i] : \text{Define} \quad \sigma(q_{it}^O) := q_{it}^A$$

Therefore,

$$ALG(\sigma(q_{it}^O)) = ALG(q_{it}^A) = OPT(q_{it}^O)$$

Since, by definition, $type(q_{it}^O) \geq slab(q_{it}^A)$, therefore,

$$\psi(type(q_{it}^O)) \leq \psi(slab(q_{it}^A))$$

$$OPT(q_{it}^O)\psi(type(q_{it}^O)) \leq ALG(\sigma(q_{it}^O))\psi(slab(\sigma(q_{it}^O)))$$

Hence for the queries $q$ mapped in first phase, we get:

$$OPT(q)\psi(type(q)) \ \leq \ ALG(\sigma(q))\psi(slab(\sigma(q)))$$

**Phase 2:** Lets call the queries from $Q^{OPT}$ which were mapped in phase 1 to be $Q1^{OPT}$. Define $Q2^{OPT} = Q^{OPT} - Q1^{OPT}$. Similarly call the queries of $Q$ to which queries from $Q^{OPT}$ got mapped in the phase 1 to be $Q1$. Define $Q2 = Q - Q1$. Now look at a query $q \in Q2^{OPT}$ s.t. $OPT(q)\psi(type(q))$ is maximum over all the queries in $Q2^{OPT}$. Now look at any query $q' \in Q2$. We shall show that $OPT(q)\psi(type(q)) \leq ALG(q')\psi(slab(q'))$. Lets say $OPT$ assigned query $q$ to the bidder $i_q$. Since in the first phase we mapped $min(|Q_{i_q}^{OPT}|, |Q_{i_q}^{ALG}|)$ queries, and $q \in Q_{i_q}^{OPT}$ was not mapped, hence $x_{i_q}$ must be equal to $|Q_i^{ALG}|$. Now, since the bidding function is decreasing, therefore wlg we can assume that $q =: \ q_{i_q(x_{i_q}+1)}^O$. When ALG was allocating $q'$, then the bid of bidder $i_q$ was $OPT(q)$. Hence by the allocation policy of the algorithm, $OPT(q)\psi(type(q)) \leq OPT(q)\psi(slab(q)) \leq ALG(q')\psi(slab(q'))$.

Therefore, in particular,

$$\forall q \in Q2^{OPT}, OPT(q)\psi(type(q)) \leq ALG(\sigma(q))\psi(slab(\sigma(q)))$$

Hence the lemma holds.

□

Let $\alpha_i^w$ be the money obtained by OPT from bidders of type $i$ with keyword $w$ only. Similarly let $\beta_i^w$ be the portion of money obtained by ALG in slab $i$ from keyword $w$ only. Now we will aggregate the above result to prove the following lemma.

**Lemma 4** $\sum_{i=1}^{k-1} \psi(i)(\alpha_i^w - \beta_i^w) \leq 0$

The proof of this lemma can be found in the appendix. Summing over all the keywords $w$, we get

**Corollary 5** $\sum_{i=1}^{k-1} \psi(i)(\alpha_i - \beta_i) \leq 0$

Now, the calculations follow as in [3]: Using Corollary 5, Equation 1 and the definition of the trade-off function $\psi$, we get that the loss of the algorithm is at most $OPT/e$, hence proving the theorem.

# 5 Discussions

## 5.1 The Difficulty with more Expressive Models

We show that providing even slightly more (non-trivial) expressiveness leads to computational issues. Consider the case of *Group Budgets*: Instead of restricting to local budget constraints on a single keyword, the bidders are allowed to set a local budget on a group of keywords.

Suppose that the set of keywords is $\{w_0,\ w_1,\ ..,\ w_k\}$. Let $c_w$ represent the number of queries of keyword $w$. Consider the following instance: There is a single bidder and his bid on all the keywords is one dollar. His budget is, lets say, $t * k$, and has following $k$ constraints on group of keywords:

$\forall i \in [1, k],\ c_{w_0} + c_{w_i} \le t$

Now consider the two sequence of queries $(w_0\ t\ times)$ and $(w_0\ t\ times, w_1\ t\ times, .., w_k\ t\ times)$. Its easy to see that: No randomized algorithm can do better than 1/2 on both the sequences. Also Greedy has a factor $1/k$ on the second sequence. Thus the extension to group budgets loses the computational possibilities available in our decreasing bids model. We believe that our bidding model achieves the correct trade-off between simplicity, expressivity and computational complexity.

## 5.2 Beyond the factor $1 - 1/e$ when bids are strictly decreasing

We now show how tightening Corollary 5 helps in getting bounds better than 1-1/e. Later we will try to see the conditions which tighten the Corollary 5.

Suppose we had that $\sum_{i=1}^{k-1} \psi(i)(\alpha_i - \beta_i) \le -x$. This $(-x)$ goes directly to the objective function of the dual LP considered in the analysis of MSVV (see[3] for details). Since the objective function of the dual represents the maximum loss of the algorithm as compared to OPT, hence the total loss becomes $\frac{OPT}{e} - x$, and the competive ratio of the algorithm will be $(1 - \frac{1}{e} + \frac{x}{OPT})$.

What are the cases when the $x$ value is substantial? We believe that one case is when the bid curves decrease rapidly. The intuition is that if for a bidder $i$ and keyword $w$, the bad case that the algorithm allocates less queries of type $w$ to $i$ than OPT does, is actually a good case. This is so because OPT derives much lesser profit for the extra queries (since they are farther in the bid-curve), while ALG allocates these elsewhere, more profitably. Characterizing this gain over $1 - 1/e$ in terms of input parameters (such as the derivative of the bid-curves) remains an open question. Similarly, we expect the performance of Greedy to be better than 1/2 in such cases.

# References


[1] N. Devanur and V. Vazirani. The spending constraint model for market equilibrium: Algorithmic, existence and uniqueness results. *STOC*, 2004.

[2] B. Kalyanasundaram and K. R. Pruhs. An optimal deterministic algorithm for online b -matching. *Theoretical Computer Science*, 233(1–2):319–325, 2000.

[3] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *FOCS*, 2005.

[4] N. Nisan. Bidding languages for combinatorial auctions. In *In Combinatorial Auctions by Cramton, Shoham and Steinberg (eds.),*, 2005.

[5] D. Parkes, T. Sandholm. Optimize-and-Dispatch Architecture for Expressive Ad Auctions. *EC*, 2005.

[6] V. Vazirani. Spending constriant utilities, with applications to the adwords market. *Manuscript*, 2006.

# A Proof of Lemma 4

**Proof :** Let $Q_w$ be the set of queries of keyword $w$. From lemma 3, we get:

$$\sum_{q \in Q_w : type(q) \leq k-1} [OPT(q)\psi(type(q)) - ALG(\sigma(q))\psi(slab(\sigma(q)))] \leq 0$$

$$\sum_{q \in Q_w : type(q) \leq k-1} OPT(q)\psi(type(q)) \leq \sum_{q \in Q_w : type(q) \leq k-1} ALG(\sigma(q))\psi(slab(\sigma(q)))$$

$$\sum_{q \in Q_w : type(q) \leq k-1} OPT(q)\psi(type(q)) \leq \sum_{q \in Q_w : type(q) \leq k-1} ALG(q)\psi(slab(q))$$

Now notice that in the MSVV algorithm, whenever a bidder hits $k^{th}$ slab, algorithm stops allocating any more queries to him, hence

$$\sum_{q \in Q_w} ALG(q)\psi(slab(q)) = \sum_{i=1}^{k-1} \sum_{q \in Q_w : slab(q)=i} ALG(q)\psi(i)$$

$$= \sum_{i=1}^{k-1} \psi(i)\beta_i^w$$

Also,

$$\sum_{q \in Q_w : type(q) \leq k-1} OPT(q)\psi(type(q)) = \sum_{i=1}^{k-1} \sum_{q \in Q_w : type(q)=i} OPT(q)\psi(i)$$

$$= \sum_{i=1}^{k-1} \psi(i)\alpha_i^w$$

□