A Cascade Model for Externalities in Sponsored Search

David Kempe*
Department of Computer Science
University of Southern California
Los Angeles, CA 90089–0781
dkempe@usc.edu

Mohammad Mahdian
Yahoo! Research
Santa Clara, CA, 95054
mahdian@alum.mit.edu

ABSTRACT

One of the most important yet insufficiently studied issues in online advertising is the externality effect among ads: the value of an ad impression on a page is affected not just by the location that the ad is placed in, but also by the set of other ads displayed on the page. For instance, a high quality competing ad can detract users from another ad, while a low quality ad could cause the viewer to abandon the page altogether.

In this paper, we propose and analyze a model for externalities in *sponsored search ads*. Our model is based on the assumption that users will visually scan the list of ads from the top to the bottom. After each ad, they make independent random decisions with ad-specific probabilities on whether to continue scanning. We then generalize the model in two ways: allowing for multiple separate blocks of ads, and allowing click probabilities to explicitly depend on ad positions as well. For the most basic model, we present a polynomial-time incentive-compatible auctions mechanism for allocating and pricing ad slots. For the generalizations, we give approximation algorithms for the allocation of ads.

1. INTRODUCTION

Online advertising auctions are run with the goal of assigning advertising slots to bidders in such a way as to maximize social welfare or the revenue of the auctioneer. The common setup is as follows: k slots are available for ads, and may be assigned to (some of) n bidders. When users *click* on an advertiser's ad, the assumption is that a certain percentage of these clicks will translate into purchases and thus revenue for the advertiser. In other words, in the type of auctions we consider here, only clicks are of interest to the bidders, as opposed to *impressions*, which would matter if the goal were to increase product awareness.

The key quantity an advertiser a is interested in with respect to slot i is the *click-through rate*, the probability that ad a, if placed in slot i, will be clicked. The larger the click-through rate, the larger the expected revenue of the advertiser. Hence, any auction aiming to maximize social welfare will need to be based on a model of click-through rates of combinations of ads and slots.

Traditional models [11, 22] are based on the assumption that the click-through rate depends solely on the relevance of the ad a and the prominence of slot i on the page. In fact, the most commonly used model makes the even stronger assumption that it is the *product* of the two quantities. The model thus completely discounts the effects of *other ads* shown on the same page. Intuition suggests that a high-quality relevant ad placed more prominently can detract from another ad. Conversely, a very low-quality ad may entice the viewer to completely disregard ads on the page. More generally, different ads on a page appear to alter each other's click-through rates.

In economics jargon, this effect is called an *externality* of an ad. Ghosh and Mahdian [12] initiated the study of externalities in online advertising. They proposed several models primarily in the context of lead generation advertising, i.e., when the publisher must select an unordered set of advertisers. The main model in [12] is based on a rational choice model for the advertising audience. However, for most of these models, the allocation problem is intractable.

In this paper, we focus exclusively on the case of *sponsored search ads*. Here, the publisher needs to select ads to be placed in a number of slots on a web page. We study the allocation problem and the design of incentive-compatible mechanisms under a simple and intuitive model called the *Cascade Model*. Our model generalizes the Cascade Model recently proposed by Craswell et al. [8] in the context of click-through rates of organic search results. (The same model is being proposed independently and simultaneously by Aggarwal et al. in this workshop [1].) The basic Cascade Model (defined formally in Section 2) assumes that the users scan through the ads in order. For each ad a, users decide probabilistically whether to click (with some ad-specific probability q_a), as well as whether to continue the scanning process, with a possibly different ad-specific probability c_a. The probabilistic continuation allows us to model the externality of prematurely terminating the scanning process as a result of either a very irrelevant ad, or a very high-quality web site leading to a purchase.

Craswell et al. [8] considered the special case of the Cascade Model where $q_a = c_a$ for all a, in the context of organic search results. Their work can be considered a follow-up on the work of Joachims et al. [14], which provides lim-

*Supported in part by NSF CAREER award 0545855, and NSF grant DDDAS-TMRP 0540420

Workshop on Ad Auctions July 8–9, 2008, Chicago, IL

ited experimental evidence for the hypothesis that the click-through rate of a search result depends on surrounding results. Craswell et al. compare the Cascade Model with four other models, including the commonly used model of separable click-through rates. They show that the Cascade Model provides the best fit to click logs of a large search engine. Since the click-through rates in organic search results and ads appear to be of a similar nature (and so far the same models have been used for both), this provides evidence that the Cascade Model can achieve a significant improvement over the currently used model of separable click-through rates.

We show that under the Cascade Model we define, the optimum allocation can be computed in polynomial time, and priced so as to lead to an incentive-compatible mechanism. We subsequently consider several generalizations of the Cascade Model. The first generalization concerns the placement of multiple separate columns or slates of ads. While each slate is scanned from top to bottom, different types of users have different orders among the slates. We show (in Section 4) that the allocation problem for this model admits a polynomial-time approximation scheme (PTAS). The second generalization (in Section 5) is a common generalization of the Cascade Model and the separable click-through rate model, which augments the Cascade Model by slot-specific click probabilities. For the allocation problem in this model, we give a simple 4-approximation algorithm as well as a quasi polynomial-time approximation scheme (QPTAS).

1.1 Related Work

Ad Auctions in general have received a lot of recent attention (see, e.g., [2, 6, 16, 20, 21]). Many of the core theoretical results (e.g., [11, 22, 2, 17]) are based on the simplifying assumption of *separable click-through rates*. That is, the probability of ad a being clicked in position i is the product $q_a \lambda_i$ of an ad-specific term and a position-dependent one. This assumption has been made mostly for simplicity; experimental studies find that there is very little evidence that separable click-through rates constitute an adequate model [14, 8, 25].

As a result, several recent papers have proposed models for externalities between ads. In addition to the work by Craswell et al. [8], in this workshop, a paper by Aggarwal et al. [1] proposes and analyzes the same Cascade Model as this paper. A paper by Das et al. [9] studies a somewhat different model of externalities. In their model, the click-through probabilities are essentially the same as in the standard separable model. However, the authors model externalities in the *conversion* of clicks, in that users will purchase from *at most one* of the sites they visited.

Athey and Ellison [4] consider a fairly general model of externalities. In their model, ads have (initially unknown) probabilities of meeting the viewer's needs, and viewers will continue clicking on ads until the need is met, or negative outcomes of previous clicks convince the user that the success probability is so low as to not be worth the cost of further clicks. If the ads are sorted by decreasing likelihood of meeting the viewer's needs, the resulting behavior is akin to the one in our model: users scan the ads from the top, and after each click make a decision on whether to continue searching. The precise condition for termination differs from the Cascade Model. Furthermore, the authors of [4] are concerned with equilibrium strategies under strong assumptions about the distributions of user costs and bids, whereas in this paper, our focus is on the allocation and mechanism design problem for *arbitrary* parameter settings.

While there has been a huge body of work in advertising on the effects of ad placement, size, content, etc. on readers' attention and recall (see, e.g., [10, 19] for related work), there appears to be — somewhat surprisingly — no study of the externalities between ads even in traditional media such as printed advertising or TV. Thus, a comparison between the Cascade Model and traditional models for externalities in advertising is not possible.

2. CLICK-THROUGH MODELS AND ALLOCATIONS

A publisher needs to choose ads from a set of n ads to display in k slots on a page, numbered sequentially from 1 to k. Slot number 1 is the top slot, and receives the maximum attention, while higher-numbered slots represent lower positions on the page which naturally receive less attention. Attention to ads is usually measured by the number of times users click on the ads (though see [10]). Each advertiser i specifies an amount b_i: how much they are willing to pay for each click on their ads. In order to optimize either his revenue or the social welfare, the publisher therefore needs to predict the probability that an ad is clicked, and take these predictions into account when allocating the slots to ads.

The *click-through rate (CTR)* of an ad is the probability that it receives a click. In principle, this probability could depend on everything on the page, including the ad itself, the position where it is placed, other ads placed in other slots, as well as seemingly less relevant other content. Since a model with so many parameters will not be useful for designing a prediction and allocation algorithm, the models currently used simplify the dependence of click-through rates on the information on the page.

The simplest model, which is currently widely used in the industry and also is the basis for most theoretical work in the area (e.g., [11, 22, 2, 17]), is based on *separable click-through rates*. It assumes that the CTR of an ad $a \in \{1, \ldots, n\}$ placed in position $i \in \{1, \ldots, k\}$ is the product $q_a \lambda_i$. Here, q_a measures the intrinsic *quality* or relevance of ad a, the probability that a user, seeing ad a, will actually click on it. λ_i measures the *prominence* of slot i, and is the probability that the user will see slot i. It is commonly assumed that λ_i is monotonically non-increasing in i. The main advantage of this model is its simplicity. Among others, simply sorting the advertisers by decreasing $b_a q_a$ yields an optimal allocation of the ad space.

2.1 The Cascade Model

One of the main drawbacks of the separable model is that it completely ignores externalities between ads. Both anecdotal evidence and user studies [14, 8] suggest that externalities are common, and that the separable model does not provide accurate fits to real world data. Therefore, we analyze a natural *Cascade Model* in this paper.

In the basic Cascade Model, each ad a, in addition to the intrinsic quality q_a, has a second parameter c_a, called its *continuation probability*. For notational convenience, we denote empty slots by $\perp$, with the understanding that $q_\perp = 0$ and $c_\perp = 1$. The model assumes that the user behaves as follows:

1. Start with the ad a_1 in slot 1.
2. When looking at the ad a_i in slot i, click on it with probability q_{a_i}.
3. Independently of whether ad a_i was clicked or not, continue to slot $i+1$ with probability c_{a_i}; otherwise, terminate the scanning process.
4. Terminate the scanning process also once no more ads remain.

Thus, if an ad a with a very small continuation probability c_a precedes another ad a', then a' is unlikely to be ever seen by the user. This models either high-quality ads which satisfy the need of the user, or off-topic ads which might frustrate the user. Under this model, assuming ads $a_1, \ldots, a_k$ are in slots $1, \ldots, k$, the click-through rate of ad a_i is

$$r_{a_i} \quad = \quad q_{a_i} \cdot \prod_{j=1}^{i-1} c_{a_j}. \tag{1}$$

Since we will frequently be considering the probability of reaching a particular slot i, we define $C_i = \prod_{j=1}^{i-1} c_{a_j}$.

REMARK 2.1. *It might seem more natural to allow for two different conditional probabilities c_a^+, c_a^-, where c_a^+ is the continuation probability if ad a is clicked, and c_a^- the continuation probability if a is not clicked. However, in that case, we could simply define $c_a = q_a c_a^+ + (1-q_a)c_a^-$; it is easy to see that the resulting click-through rates under c_a^+ and c_a^- are exactly those for c_a according to Equation (1). Since the click-through rates for all positions are all that matters for the model and its analysis, it suffices to focus on the simpler model we defined above.*

As we mentioned above, a simpler version of the Cascade Model has been proposed recently by Craswell et al. [8] in the context of organic search results. They assume, perhaps somewhat unrealistically, that $c_a = q_a$ for all ads a. Furthermore, in the context of organic search, there is no *willingness-to-pay* parameter b_a.

2.2 Generalized Cascade Models

In the simple Cascade Model above, the decay of the click-through rates as a function of slot positions depends solely on the ads that are placed in the slots, but not the slots themselves. Furthermore, all users are assumed to be identical, and scan the ads in the same order (e.g., from top to bottom). In practice, both assumptions are probably overly simplistic. We elaborate on two generalized models below.

2.2.1 Multiple Ad Slates

Many search engines present sponsored search ads in multiple different slates, e.g., some preceding the organic search results and some on the right-hand side. As a result, different users may have different orders in which they scan the ads. We will first define a general *Permuted Cascade Model*, and then focus on the special case in which there are a constant number of slates, each of which is always scanned from top to bottom.

In the Permuted Cascade Model, for each permutation π of $\{1, \ldots, k\}$, a fraction f_π of users will scan the ads in the order $\pi(1), \pi(2), \ldots, \pi(k)$. (Thus, $\sum_\pi f_\pi = 1$.) If the ad assignment is a_i, then a user with scanning order π will have a click-through rate of

$$r_{a_i}^{(\pi)} \quad = \quad q_{a_i} \cdot \prod_{j=1}^{\pi^{-1}(i)-1} c_{a_{\pi(j)}} \tag{2}$$

for the ad a_i in slot i. Similarly to the simple Cascade Model, we write $C_i^{(\pi)} = \prod_{j=1}^{\pi^{-1}(i)-1} c_{a_{\pi(j)}}$ for the probability that the user will look at slot i. The overall click-through rate of ad a_i in slot i is thus

$$r_{a_i}^{(f)} \quad = \quad \sum_\pi q_{a_i} f_\pi C_i^{(\pi)}; \tag{3}$$

the objective is to find an allocation maximizing $\sum_i r_{a_i}^{(f)} b_{a_i}$.

Here, we are particularly interested in the special case in which there is a constant number s of *slates*. Slate i has k_i slots, and each user scans each slate from top to bottom (until stopping the scan). However, different users might have different orders over the slates. Formally, the only permutations π with non-zero frequencies f_π are those defined in terms of a permutation $\psi : \{1, \ldots, s\} \to \{1, \ldots, s\}$ of the slates, which induces the permutation π placing all $k_{\psi(j)}$ slots of slate $\psi(j)$, in their natural order, before all $k_{\psi(j+1)}$ slots of slate $\psi(j+1)$, for each j. We call this model the *Slated Cascade Model*. In the Slated Cascade Model, we use (j, i) to denote the i^{th} slot of the j^{th} slate, and $a_{j,i}$ to denote the ad in slot i of slate j. Furthermore, we write $C_{j,i} = \prod_{h=1}^{i-1} c_{a_{j,h}}$ for the probability of reaching slot i, given that slate j is entered, $\overline{C}_j = \prod_{i:a_{j,i} \neq \perp} c_{a_{j,i}}$ for the probability of moving on to the next slate given that slate j is entered, and

$$\hat{C}_{j,i} \quad = \quad \textstyle\sum_\psi f_\psi \cdot C_{j,i} \cdot \prod_{h=1}^{\psi^{-1}(j)-1} \overline{C}_h$$

for the overall probability of seeing slot (j, i).

In this paper, we will only focus on the Slated Cascade Model, rather than the general Permuted Cascade Model. While studying the Permuted Cascade Model may be quite interesting from a theoretical point of view (for instance, the allocation problem appears to become significantly more complex), an applicability to sponsored search is rather tenuous.

2.2.2 Position-Dependent Multipliers

A second simplifying assumption is that the click-through rates depend *only* on the continuation probabilities, but not on the actual slot in which an ad appears. In practice, it appears that users have been conditioned to assume that results listed in higher positions may be more relevant. Therefore, click-through rates will also depend on the position. To model this dependency, we define a model we call the *Cascade Model with Position-Dependent Multipliers* (CMPDM).

In the CMPDM, each slot i also has a parameter λ_i, the slot-specific probability of reading an ad in that slot, subject to scanning all the way to the slot. In accordance with most of the literature on position auctions, we assume here that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$. The expression (1) for the click-through rate is now

$$r'_{a_i} \quad = \quad \lambda_i q_{a_i} \cdot \prod_{j=1}^{i-1} c_{a_j}. \tag{4}$$

Thus, the CMPDM is a common generalization of both the Cascade Model and the separable click-through rate model, and is therefore more expressive than both. We will study the allocation problem for this model in Section 5.

2.3 Slot Allocation and Incentive-Compatibility

Given the above Cascade Model for click-through rates, the publisher needs to solve the following optimization problem in order to maximize the value. Assume, without loss of generality, that $n \geq k$. The objective is to select $\ell \leq k$ distinct ads $a_1, \ldots, a_\ell$ to maximize

$$\sum_{i=1}^{\ell} b_{a_i} q_{a_i} \cdot \prod_{j=1}^{i-1} c_{a_j}. \tag{5}$$

In the next section, we will describe an efficient algorithm for solving this optimization problem. However, solving the optimization problem requires knowledge of the parameters b_a, q_a, c_a for each bidder a. While the probabilities q_a and c_a can be learned from click-through histories, the willingness to pay is the valuation v_a which an advertiser assigns to clicks, and thus intrinsically private information. In particular, utility-maximizing advertisers may submit bids $b_a \neq v_a$ if doing so stands to improve their utilities.

In order to extract *truthful* bids from the advertisers, i.e., entice them to submit $b_a = v_a$, the publisher can charge them prices p_a per click, which may differ from the submitted bids. If charged p_a, an advertiser's utility is $r_a(v_a - p_a)$, where r_a is the click-through rate for advertiser a, and may depend on the entire assignment. A *mechanism* consists of both an allocation rule and a payment rule, giving the payments p_a per click. It is *truthful* or *incentive compatible* if the best strategy of each advertiser a, independent of the strategies of other advertisers, is to bid $b_a = v_a$. We will discuss truthful mechanisms in the Cascade Model below.

REMARK 2.2. *In the discussion above, we are assuming that the value per click for an advertiser is not affected by the set of other ads on the page. While one can imagine scenarios where this assumption is violated, for instance in the presence of spiteful bidding [5, 18], the effect of other ads on the* value *of a click for ad a appears to be second-order. Furthermore, taking this effect into account would be difficult in practice, as it involves changing the bidding language to allow advertisers to express valuations contingent on other ads displayed on the page.*

3. WINNER DETERMINATION IN THE CASCADE MODEL

In this section, we show that the optimal allocation for the simple Cascade Model can be computed by a dynamic program. The key tool for deriving this program is a lemma showing that whichever ads are shown must follow a simple ordering. The results of this section were obtained independently by Aggarwal et al. [1], in a paper appearing in the same workshop as this one.

LEMMA 3.1. *Assume that the optimal solution places ad a_i in position i. Then, w.l.o.g.,*

$$\frac{b_{a_1} q_{a_1}}{1-c_{a_1}} \geq \frac{b_{a_2} q_{a_2}}{1-c_{a_2}} \geq \cdots \geq \frac{b_{a_k} q_{a_k}}{1-c_{a_k}}. \tag{6}$$

Proof. The proof of this lemma relies on an exchange argument similar to the types of arguments in the analysis of greedy scheduling algorithms (see, e.g., [15]).

Assume that there is a position $i < k$ such that $\frac{b_{a_i} q_{a_i}}{1-c_{a_i}} < \frac{b_{a_{i+1}} q_{a_{i+1}}}{1-c_{a_{i+1}}}$. Let $a = a_i, a' = a_{i+1}$ be the two ads in those positions, and consider the alternative ordering placing a in position $i+1$, and a' in position i, while leaving all other ads in the same slots as before. By Equation (1), the click-through rates for all positions $j \notin \{i, i+1\}$ remain the same. Recalling that $C_i = \prod_{j=1}^{i-1} c_{a_j}$, the new click-through rates for ads a and a' are $r'_a = q_a c_{a'} \cdot C_i$ and $r'_{a'} = q_{a'} \cdot C_i$. Thus, the total change in value is

$$\begin{aligned} & r'_a b_a + r'_{a'} b_{a'} - (r_a b_a + r_{a'} b_{a'}) \\ = \;& C_i (b_{a'} q_{a'} (1 - c_a) - b_a q_a (1 - c_{a'})) \\ \geq \;& C_i (b_a q_a (1 - c_{a'}) - b_a q_a (1 - c_{a'})) \\ = \;& 0, \end{aligned}$$

where the second to last step followed from the assumption about the ordering. Hence, swapping a and a' will not decrease the value. By repeating such swaps inductively, we can eventually obtain a solution of at least the same value, and sorted according to (6). ■

Using the above lemma, we can use dynamic programming to design a polynomial-time algorithm for the winner determination problem. First, all ads are sorted in non-increasing order of their ratios $\frac{b_a q_a}{1-c_a}$, so that

$$\frac{b_1 q_1}{1-c_1} \geq \frac{b_2 q_2}{1-c_2} \geq \cdots \geq \frac{b_n q_n}{1-c_n}.$$

We then fill out a dynamic programming table $A_{n \times k}$, whose entry $A[a, i]$ contains the optimum value that can be obtained from ads $a, \ldots, n$ in positions $i, \ldots, k$, *conditioned* on the ad in slot i being read. Once this table is filled out, the solution of the problem is contained in the entry $A[1, 1]$. To fill this table, we use the following recurrence:

$$A[a, i] = \max(A[a+1, i], b_a q_a + c_a A[a+1, i+1]).$$

This recurrence captures that if ad a is placed in position i, then its *conditional expected value* is $b_a q_a$, and the reader will continue to slot $i+1$ with probability c_a. Thus, the expected conditional value obtained from slots $i+1, \ldots, k$ is $c_a A[a+1, i+1]$, since the ads in slots $i+1, \ldots, k$ will be chosen optimally as well. The important role of Lemma 3.1 is that it ensures we are not missing the optimum solution by always placing ad a in a slot before ads $a+1, \ldots, n$. Summing up, we obtain the following theorem.

THEOREM 3.2. *There is an algorithm with a running time of $O(n \log n + nk)$ which computes the optimal placement of n ads in k slots in the simple Cascade Model.*

3.1 Incentive-Compatible Mechanism Design

To turn the above algorithm into an incentive compatible mechanism, we can use a pricing scheme based on the classical Vickrey-Clarke-Groves (VCG) mechanism [24, 7, 13]. The VCG payment scheme charges each bidder a an amount equal to the externality this bidder imposes on other bidders. The externality can be calculated by removing a from the set of advertisers, running the algorithm again, and computing the total utility of all advertisers in the resulting solution. The VCG payment is then the difference between the value of all advertisers in this new solution, and the value of all advertisers except a in the original optimum.

It is well known, and not very hard to see, that this payment scheme gives an incentive-compatible, efficient mechanism for allocating and pricing ads in the simple Cascade Model. Computing the prices can be accomplished with k separate invocations of the dynamic program described above, for a total running time of $O(n \log n + nk^2)$.

4. MULTIPLE AD SLATES

In this section, we study the Slated Cascade Model, in which there is a constant number s of slates, with slate j having k_j slots. For this scenario, we give a polynomial-time approximation scheme (PTAS). Thus, we rule out that the problem is APX-hard. However, at this point, it is open whether the problem with multiple slates is NP-hard, or could be solved optimally in polynomial time.

The algorithm first discretizes some of the parameters. It then exhaustively searches over all possible aggregate continuation probabilities in each slate, and then runs a dynamic program to assign ads to slots. By conditioning on the aggregate continuation probabilities of slates, the choices for slates become de facto independent. Throughout, we let δ be a suitably small constant whose precise value will be determined below.

4.1 Ignoring Small Probabilities

In order to show that we can ignore, at the cost of only a small loss in approximation guarantee, any ads that will be seen only with small probability, we first prove the following lemma. We prove it in a fairly general form with position dependent multipliers and multiple slates, since it will also be a key building block for our approximation algorithms in Section 5. Recall from Section 2.2.1 that we use (j, i) to denote the i^{th} slot in the j^{th} slate, $C_{j,i}$ to denote the probability of reaching slot i in slate j, given that the user starts scanning slate j at all, and $\hat{C}_{j,i}$ for the overall probability of ever seeing slot (j, i).

LEMMA 4.1. *Let the position dependent multipliers of slate j be $\lambda_{j,1} \geq \lambda_{j,2} \geq \cdots \geq \lambda_{j,k_j}$. Let ψ be any distribution over permutations of slates, and* OPT *the value of the optimum solution.*

For any $\delta > 0$, there is a solution (possibly leaving some slots empty) of value at least $(1 - \delta) \cdot \mathrm{OPT}$ such that for all non-empty slots j, we have $C_{j,i} \geq \delta$. That is, each non-empty slot is reached with probability at least δ, given that its slate is entered in the first place.

Proof. Let $(a_{j,i})_{j,i}$ be an optimal solution of value OPT. For every slate j, let $r_j \leq k_j$ be the last slot containing an ad, and ℓ_j the largest index such that $C_{j,\ell_j} \geq \delta$. Consider the solution that is obtained by moving the ads $a_{j,\ell+1}, \ldots, a_{j,r_j}$ to slots $(j, 1), \ldots, (j, r_j - \ell_j)$, for all j simultaneously, while leaving the remaining slots empty.

Let $C'_{j,i}$ be the new probability of seeing slot (j, i) after the change, given that slate j is scanned. By the choice of ℓ_j, we immediately obtain that $C'_{j,i-\ell_j} \geq \frac{1}{\delta} C_{j,i}$ for all $i \geq \ell_j + 1$. That is, the ad formerly in position (j, i) for $i \geq \ell_j + 1$ is now at least $\frac{1}{\delta}$ times as likely to be reached, given that slate j is scanned. Furthermore, because we only removed ads from slates, we immediately have that $\overline{C}'_j \geq \overline{C}_j$ for all slates j, i.e., it only becomes more likely that scanning of any slate j will finish. Thus, with $\hat{C}'_{j,i}$ denoting the new overall probability of seeing slot (j, i), we obtain that $\hat{C}'_{j,i-\ell_j} \geq \frac{1}{\delta} \hat{C}_{j,i}$, i.e., each remaining ad is at least $\frac{1}{\delta}$ times as likely to be seen after the change.

Each such ad is now in a slot (j, i) whose position dependent multiplier $\lambda_{j,i}$ is at least as large as the original one, by the sorting of the multipliers. The value of the new solution is thus

$$\begin{aligned} & \sum_j \sum_{i=1}^{r_j - \ell_j} \lambda_{j,i} \hat{C}'_{j,i} q_{a_{j,i+\ell_j}} b_{a_{j,i+\ell_j}} \\ \geq\; & \frac{1}{\delta} \sum_j \sum_{i=\ell_j+1}^{r_j} \lambda_{j,i} \hat{C}_{j,i} q_{a_{j,i}} b_{a_{j,i}}, \end{aligned}$$

where we used the above argument as well as the fact that the $\lambda_{j,i}$ are sorted. Because the new solution cannot be better than OPT, we obtain that

$$\sum_j \sum_{i=\ell_j+1}^{r_j} \lambda_{j,i} \hat{C}_{j,i} q_{a_{j,i}} b_{a_{j,i}} \quad \leq \quad \delta \cdot \mathrm{OPT},$$

and thus

$$\sum_j \sum_{i=1}^{\ell_j} \lambda_{j,i} \hat{C}_{j,i} q_{a_{j,i}} b_{a_{j,i}} \quad \geq \quad (1 - \delta) \cdot \mathrm{OPT}.$$

In other words, removing the ads in positions $\ell_j + 1, \ldots, r_j$ cannot decrease the value by more than $\delta \cdot \mathrm{OPT}$, and ensures that $C_{j,i} \geq \delta$ for all non-empty slots (j, i) by construction. ■

4.2 Description of the Algorithm

We are now ready to describe and analyze the steps of the algorithm in detail:

1. **Ignoring small continuation probabilities.** First, we round any continuation probability c_a that is less than $\delta/(s + 1)$ down to zero. Effectively, this ignores the value of any ad that comes after an ad with such small c_a, in the same slate j. It also changes the probability of leaving slate j to 0. We argue that this decreases the value of the solution by at most $\delta \cdot \mathrm{OPT}$. By Lemma 4.1, there is a solution of value at least $(1 - \frac{\delta}{s+1}) \cdot \mathrm{OPT}$ such that no ad follows any such low-probability ad in the same slate. For this altered solution, changing the continuation probability of $c_{a_{j,i}}$ to 0 does not affect the value of slate j.

   Now consider the impact of the modified probabilities $\overline{C}_j$ of leaving slates $j = 1, \ldots, s$. That is, for some slates, we replace $\overline{C}_j \leq \frac{\delta}{s+1}$ by $\overline{C}'_j = 0$. For each slate j, the probability of reaching it decreases by at most $\frac{\delta}{s+1}$. Let V_j be the expected value of slate j, conditioned on reaching it in the scanning process. By leaving all other slates empty, we could make sure to reach slate j with probability 1, and since the optimum solution must be at least as good, we obtain that $V_j \leq$ OPT. Summing up over all slates, the total expected value decreases by at most $\sum_j \frac{\delta}{s+1} V_j \leq \frac{s\delta}{s+1} \cdot \mathrm{OPT}$. Thus, in total, the value decreased by at most $\delta \cdot \mathrm{OPT}$.

2. **Rounding continuation probabilities.** Next, we round down each non-zero continuation probability to the nearest power of $(1 - \delta/k)$. For any slot (j, i), the resulting probability of reaching (j, i) is not changed by more than a factor of $(1 - \delta/k)^k \geq 1 - \delta$. Therefore,

this stage decreases the value of the optimal solution by at most $\delta \cdot \text{OPT}$. Subsequently, the *product* of the continuation probabilities of any subset of at most k ads is one of the $O(k^2)$ values

$$\{0, 1, (1-\delta/k)^1, \ldots, (1-\delta/k)^{k\lceil \log_{(1-\delta/k)}(\delta/(s+1))\rceil}\}.$$

Denote this set of values by $\mathcal{C}$.

3. **Enumerating over all slate probabilities.** The algorithm exhaustively enumerates all combinations of probabilities $\overline{C}_s \in \mathcal{C}$ for slates s. We call such an s-tuple $(\overline{C}_1, \ldots, \overline{C}_s)$ a *configuration*. Notice that there are only $O(k^{2s})$, i.e., polynomially many, configurations. So long as for each slate s, the product of the continuation probabilities is exactly $\overline{C}_s$, this enumeration makes it possible to evaluate precisely the click-through rate of any ad in any position. In particular, we use γ_s to denote the overall probability of entering slate s. Notice that all γ_s can be computed efficiently from the distribution f_ψ over permutations and the $\overline{C}_s$ values.

4. **Dynamic programming solution.** For each configuration $(\overline{C}_1, \ldots, \overline{C}_s)$, we find the optimal solution consistent with that configurations using dynamic programming. By Lemma 3.1, the advertisers in each slate must be ordered in decreasing order of their ratios $\frac{b_a q_a}{1-c_a}$. The dynamic programming idea is essentially an s-dimensional Knapsack program. First sort the ads such that $\frac{b_1 q_1}{1-c_1} \geq \frac{b_2 q_2}{1-c_2} \geq \cdots \geq \frac{b_n q_n}{1-c_n}$. The dynamic programming table has entries

$$A[a, i_1, \ldots, i_s, y_1, \ldots, y_s]$$

for all $1 \leq a \leq n, 0 \leq i_j \leq k_j, y_j \in \mathcal{C}$, and $\overline{C}_j \leq y_j \leq 1$ for every slate j. This entry contains the optimal total value that can be obtained from the last i_j slots of each slate $j = 1, \ldots, s$, where all ads are from the set $a, \ldots, n$, assuming that the product of the continuation probabilities of the first $k_j - i_j$ slots of slate j is y_j.

This entry of the dynamic programming table can be computed by considering all options for ad a. By Lemma 3.1, ad a is either not used at all, or is placed at the first “empty” slot of one of the slates. If ad a is not used at all, then the optimum value is just $A[a+1, i_1, \ldots, i_s, y_1, \ldots, y_s]$. Otherwise, if ad a is used in slate j (which is possible only if $i_j > 0$), the optimum value is

$$A[a+1, i_1, \ldots, i_j - 1, \ldots, i_s, y_1, \ldots, y_j \cdot c_a, \ldots, y_s] + \gamma_j y_j b_a q_a.$$

The optimum is then simply the maximum over these $s+1$ different options, and can be computed in constant time $O(s)$. Overall, the dynamic program then takes time $O(nk^{3s})$.

The rounding stages in the above algorithm lose at most $2\delta \cdot \text{OPT}$ of the value of the optimal solution. The last two stages use $O(k^{2s})$ invocations of a dynamic program, each of which takes time $O(nk^{3s})$, to compute the optimal solution for the rounded instance. Hence, by taking $\delta = \epsilon/2$, we obtain

Theorem 4.2. *For every constant $\epsilon > 0$, there is a polynomial-time algorithm for the winner determination problem in the Slated Cascade Model which always outputs a solution whose value is at least $(1-\epsilon) \cdot \text{OPT}$.*

5. POSITION-DEPENDENT MULTIPLIERS

We next investigate the allocation problem for the Cascade Model with Position-Dependent Multipliers (CMPDM) of Section 2.2. Recall that the click-through rate r'_{a_i} now also has a factor of λ_i, a slot specific click probability, in addition to the product of the preceding continuation probabilities.

The main difficulty in designing an algorithm for the winner determination problem in the CMPDM is that the equivalent of Lemma 3.1 no longer holds. Intuitively, high continuation probabilities are much less important if subsequent slots have very low slot-specific λ_i values. Thus, there can be no simple sorting criterion based solely on properties of the ads themselves.

In this section, we first give a simple polynomial-time 4-approximation algorithm in the CMPDM. We then describe a more complicated (and slower) quasi-polynomial-time approximation scheme (QPTAS), based on the ideas underlying the PTAS in Section 4. Our QPTAS will run in time $O(n \cdot 2^{O(\log^2 k)})$ for any fixed ϵ, while achieving an approximation factor of $1-\epsilon$. While we present approximation algorithms in this section, it is currently open whether the allocation problem in the CMPDM is NP-hard, or can be solved optimally in polynomial time.

5.1 A 4-approximation algorithm

Our 4-approximation algorithm is based on a reduction to the Knapsack problem. First, by applying Lemma 4.1 with $\delta = \frac{1}{2}$ (and just one slate), we can restrict our attention to solutions where $C_i \geq \frac{1}{2}$ for all i except possibly the last ad, so long as we are willing to incur a factor 2 in the approximation guarantee. For such solutions, the value of any selected ad a_i can be easily 2-approximated, as $\frac{1}{2} \leq C_i \leq 1$. In particular, the objective function (5) can be 2-approximated by $\sum_i \lambda_i b_{a_i} q_{a_i}$.

Thus, incorporating the constraint that $C_i \geq \frac{1}{2}$ for all non-empty slots i, we obtain a Knapsack-like problem as follows. First, we can exhaustively try all ads $\hat{a}$ which will occupy the last assigned slot ℓ. The ad $\hat{a}$ will be the only one with no restrictions on its continuation probability $c_{\hat{a}}$. Because the slots are ordered by decreasing λ_i, the selected ads $a_1, a_2, \ldots, a_{\ell-1}$ will necessarily satisfy $b_{a_1} q_{a_1} \geq b_{a_2} q_{a_2} \geq \cdots \geq b_{a_{\ell-1}} q_{a_{\ell-1}}$, as can be seen by a simple exchange argument.

The optimization problem is thus to select $a_1, \ldots, a_{\ell-1}$ to maximize $\sum_{i=1}^{\ell-1} \lambda_i b_{a_i} q_{a_i} + \lambda_\ell b_{\hat{a}} q_{\hat{a}}$, subject to the constraints that $\sum_{i=1}^{\ell-1} \log_2 \frac{1}{c_{a_i}} \leq 1$ and $\ell \leq k$. This is exactly a Knapsack problem with cardinality and size constraints. With a simple generalization of the standard fully polynomial-time approximation scheme (FPTAS) for Knapsack [23], we obtain an FPTAS for this problem, and thus a $(4+\epsilon)$ approximation for the ad allocation problem, for any $\epsilon > 0$.

5.2 A quasi-PTAS

In this section, we describe a quasi-polynomial-time approximation scheme (QPTAS) for the winner determination problem in the CMPDM. The algorithm is based on very similar ideas as the one for the Slated Cascade Model. Instead of multiple slates, it divides the slots into multiple *segments*, each consisting of slots with (approximately) the same λ_i values. Within those segments, we can then use the same Dynamic Programming ideas as with the multiple slates. In order to achieve the desired approximation guarantees, the number of segments has to be $\Theta(\log k)$. Hence, the running time will end up being $O(n \cdot k^{O(\log k)})$. Again, we let δ be a suitably small constant whose precise value will be determined below. We assume without loss of generality that the largest position-dependent multiplier is $\lambda_1 = 1$ (otherwise, we could rescale all multipliers by λ_1).

1. **Removing slots with small λ_i.** As a first step, we remove all slots i with $\lambda_i < \delta/k$. Because the ad with the largest expected value $b_a q_a$ could be placed in the first slot for expected value $b_a q_a$, we know that $b_a q_a \leq \text{OPT}$ for all ads a. In particular, the expected value from any slot i with $\lambda_i < \delta/k$ will be $C_i \lambda_i b_{a_i} q_{a_i} \leq \frac{\delta}{k}\text{OPT}$. Thus, removing *all* such slots can reduce the total value by at most $\delta \cdot \text{OPT}$.

2. **Rounding position-dependent multipliers.** As a second rounding step, we round all of the remaining λ_i values down to the nearest power of $(1-\delta)$. This rounding guarantees that we lose at most a factor $C_i \delta b_{a_i} q_{a_i}$ in position i, and summing up gives us that the achievable optimum decreases by at most $\delta \cdot \text{OPT}$. After the rounding, each λ_i is one of only $O(\log k)$ distinct values $1, (1-\delta)^1, \ldots, (1-\delta)^{\lceil \log_{1-\delta}(\delta/k) \rceil}$. We call the set of all slots i with multipliers $\lambda_i = (1-\delta)^j$ the j^{th} *segment* of slots, and use $s = 1 + \lceil \log_{1-\delta}(\delta/k) \rceil$ to denote the total number of such segments, and k_j for the number of slots in segment j.

3. **Rounding continuation probabilities.** As a next step, we round the continuation probabilities in a very similar way to the rounding in the PTAS from Section 4. First, we round any continuation probability c_a that is less than δ down to zero. By Lemma 4.1, this changes the achievable optimum by at most $\delta \cdot \text{OPT}$. For notice that the lemma guarantees a solution of value at least $(1-\delta) \cdot \text{OPT}$ such that all non-empty slots i have $C_i \geq \delta$. In particular, in that solution, no ads can come after any ad a with $c_a < \delta$, so that the value is not changed at all by changing c_a to 0.

   As a final rounding step, we round down each nonzero continuation probability to the nearest power of $1 - \delta/k$. For any slot, the click-through rate changes by at most a factor of $(1-\delta/k)^{k-1} \geq 1-\delta$ as a result. In total, this stage decreases the value of the optimal solution by at most $2\delta \cdot OPT$. The rounding guarantees that the product of the continuation probabilities of any subset of at most k ads is one of the $O(k^2)$ values $0, 1, (1-\delta/k)^1, \ldots, (1-\delta/k)^{k\lceil \log_{(1-\delta/k)}(\delta) \rceil}$. Denote this set of values by $\mathcal{C}$.

4. **Enumerating and Dynamic Programming.** The remainder of the solution is nearly identical to the approach taken for the Slated Cascade Model. We can regard each segment like a slate before, and exhaustively try all corresponding s-tuples of $\overline{C}_j$ values. The only change is that the multipliers for each segment j are simply the corresponding λ_i values, i.e., $(1-\delta/k)^j$, rather than the somewhat more cumbersome γ_j values. Because there are now $s = \Theta(\log k)$ slates, the dynamic program has to be over $(2s+1)$-tuples of one index a of an ad, s indices of positions in segments (with at most k possible values), and s probabilities which are products of continuation probabilities (with at most $O(k^2)$ possible values). Hence, the running time is $O(nk^{3s}) = O(n2^{O(\log^2 k)})$. Thus, it is quasi-polynomial for any fixed δ.

The rounding stages in the above algorithm lose at most a total of $4\delta \cdot \text{OPT}$ of the value of the optimal solution. The last stage computes the optimal solution for the rounded instance with $O(2^{O(\log^2 k)})$ invocations of a dynamic program with running time $O(n \cdot 2^{O(\log^2 k)})$. Hence, by taking $\delta = \epsilon/4$, we obtain

THEOREM 5.1. *For every constant $\epsilon > 0$, there is a quasi-polynomial-time algorithm for the winner determination problem in the CMPDM which always outputs a solution whose value is at least $(1-\epsilon) \cdot \text{OPT}$.*

REMARK 5.2. *The complexity of the QPTAS is considerably higher than the 4-approximation algorithm described earlier. In a practical application, it might be more reasonable to use the 4-approximation algorithm (perhaps with a parameter other than $\frac{1}{2}$ to achieve a better performance in practice). The QPTAS both clarifies the theoretical complexity of the problem, and can also give insights into techniques like rounding that can be used in practice.*

5.3 Incentive compatible implementation.

The VCG payment scheme used in Section 3.1 relies on an algorithm that finds the *exact* optimal outcome. Therefore, an approximation algorithm for the allocation problem does not automatically translate into an incentive-compatible mechanism. However, since the agents' types are one-dimensional, a generalization of the VCG result (see, for example, [3]) applies to this setting. This generalization of VCG shows that if an allocation algorithm is *monotone* (i.e., increasing the bid b_a can never decrease an agent's likelihood of winning), then it can be truthfully implemented. Fortunately, both our 4-approximation algorithm and our QPTAS only round continuation probabilities and do not round the values. Therefore, it is not hard to show that they are monotone, and can be truthfully implemented. The only difficulty is that it is not clear that the payments that guarantee a truthful implementation can be computed in polynomial time.

6. DISCUSSION AND FUTURE RESEARCH DIRECTIONS

In this paper, we studied the Cascade Model for externalities in sponsored search advertising. In comparison with the model defined in [12], the main advantage of our model is that it is tractable, i.e., the allocation problem can be solved efficiently. At the same time, the experimental results of Craswell et al. [8] show that even when the two parameters q_a and c_a of each agent are collapsed into one by setting $q_a = c_a$, the model still gives a significant improvement over the separable CTR model that is currently used. Thus, the model is particularly appealing for practical purposes.

One main shortcoming of our model is that it does not address situations where the search term is ambiguous (e.g., "Apple" which can be matched to ads on Apple computers and on the fruit). In such cases, the set of candidate ads consists of multiple classes with significantly different "inter-class" and "intra-class" externalities. However, such situations are rare and likely to become more so with the advance of new search technologies aimed at modeling intent.

As for future research, one of the important problems we are currently investigating is to develop CTR-learning algorithms that can learn the parameters of our model, and pursue an exploration-exploitation strategy that converges to the optimal solution over time. It would also be desirable to determine the complexity of computing an *exact* solution in the Slated Cascade Model or in the Cascade Model with Position-Dependent Multipliers. While we presented a PTAS resp. a QPTAS for these models, we do not currently know whether these problems are NP-hard to solve optimally.

Acknowledgments

We would like to thank David Pennock for suggesting the generalization to the model with position-dependent multipliers, the authors of [1] and [9] for sharing preprints of their papers, and several anonymous reviewers for useful feedback.

7. REFERENCES

[1] G. Aggarwal, J. Feldman, S. Muthukrishnan, and M. Pál. Sponsored search auctions with markovian users. In *Proc. 4th Workshop on Ad Auctions*, 2008.

[2] G. Aggarwal, A. Goel, and R. Motwani. Truthful auctions for pricing search keywords. In *Proc. 8th ACM Conf. on Electronic Commerce*, pages 1–7, 2006.

[3] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. 42nd IEEE Symp. on Foundations of Computer Science*, pages 482–491, 2001.

[4] S. Athey and G. Ellison. Position auctions with consumer search. Working Paper, September 2007.

[5] F. Brandt, T. Sandholm, and Y. Shoham. Spiteful bidding in sealed-bid auctions. In *Proc. 22nd Intl. Joint Conf. on Artificial Intelligence*, pages 1207–1214, 2007.

[6] M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, A. Karlin, C. Mathieu, and M. Schwarz. Greedy bidding strategies for keyword auctions. In *Proc. 9th ACM Conf. on Electronic Commerce*, pages 262–271, 2007.

[7] E. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[8] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proc. 1st Intl. Conf. on Web Search and Data Mining*, 2008.

[9] A. Das, I. Giotis, A. Karlin, and C. Mathieu. On the effects of competing advertisements in keyword auctions. Unpublished Manuscript, May 2008.

[10] X. Drèze and F.-X. Hussherr. Internet advertising: Is anybody watching? *J. of Interactive Marketing*, 17:8–23, 2003.

[11] 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*, 97:242–259, 2007.

[12] A. Ghosh and M. Mahdian. Externalities in online advertising. In *17th International World Wide Web Conference*, 2008.

[13] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[14] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2), 2007.

[15] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison-Wesley, 2005.

[16] S. Lahaie. An analysis of alternative slot auction designs for sponsored search. In *Proc. 8th ACM Conf. on Electronic Commerce*, pages 218–227, 2006.

[17] S. Lahaie, D. Pennock, A. Saberi, and R. Vohra. Sponsored search auctions. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 28. Cambridge University Press, 2007.

[18] L. Liang and Q. Qi. Cooperative or vindictive: Bidding strategies in sponsored search auctions. In *Proc. 3rd Workshop on Internet and Network Economics (WINE)*, pages 167–178, 2007.

[19] G. Lohse. Consumer eye movement patterns on yellow pages advertising. *J. of Advertising*, 26:61–73, 1997.

[20] M. Mahdian, H. Nazerzadeh, and A. Saberi. Allocating online advertisement space with unreliable estimates. In *Proc. 9th ACM Conf. on Electronic Commerce*, pages 288–294, 2007.

[21] S. Muthukrishnan, M. Pál, and Z. Svitkina. Stochastic models for budget optimization in search-based advertising. In *Proc. 3rd Workshop on Internet and Network Economics (WINE)*, pages 131–142, 2007.

[22] H. Varian. Position auctions. *Intl. J. of Industrial Organization*, 25:1163–1178, 2007.

[23] V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[24] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *J. of Finance*, 16:8–37, 1961.

[25] J. Wortman, Y. Vorobeychik, L. Li, and John Langford. Maintaining equilibria during exploration in sponsored search auctions. In *Proc. 3rd Workshop on Internet and Network Economics (WINE)*, pages 119–130, 2007.