Pricing Guaranteed Contracts in Online Display Advertising

Vijay Bharadwaj
Yahoo! Inc.
Santa Clara, CA, USA
vbharadw@yahoo-inc.com

Wenjing Ma
Yahoo! Inc.
Santa Clara, CA, USA
wenjingm@yahoo-inc.com

Michael Schwarz
Yahoo! Research
Santa Clara, CA, USA
mschwarz@yahoo-inc.com

Jayavel Shanmugasundaram
Yahoo! Research
Santa Clara, CA, USA
jaishan@yahoo-inc.com

Erik Vee
Yahoo! Research
Santa Clara, CA, USA
erikvee@yahoo-inc.com

Jack Xie*
Rich Relevance
San Francisco, CA, USA
jack@richrelevance.com

Jian Yang
Yahoo! Labs
Santa Clara, CA, USA
jianyang@yahoo-inc.com

ABSTRACT

We consider the problem of pricing guaranteed contracts in online display advertising. This problem has two key characteristics that when taken together distinguish it from related offline and online pricing problems: (1) the guaranteed contracts are sold months in advance, and at various points in time, and (2) the inventory that is sold to guaranteed contracts — user visits — is very high-dimensional, having hundreds of possible attributes, and advertisers can potentially buy any of the very large number (many trillions) of combinations of these attributes. Consequently, traditional pricing methods such as real-time or combinatorial auctions, or optimization-based pricing based on self- and cross- elasticities are not directly applicable to this problem. We hence propose a new pricing method, whereby the price of a guaranteed contract is computed based on the prices of the individual user visits that the contract is expected to get. The price of each individual user visit is in turn computed using historical sales prices that are negotiated between a sales person and an advertiser, and we propose two different variants in this context. Our evaluation using real guaranteed contracts shows that the proposed pricing method is accurate in the sense that it can effectively predict the prices of other (out-of-sample) historical contracts.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision Support*

General Terms

Algorithms, Economics

Keywords

Pricing, Display Advertising

1. INTRODUCTION

We propose a method for pricing guaranteed contracts in online display advertising. In guaranteed display advertising, advertisers can buy contracts that specify targeted user visits (e.g., Males in California who visit Sports pages), a future duration for the contract (e.g., October - December 2010), and the number of user visits they are interested in obtaining (e.g., 100 million), and Web publishers *guarantee* these contracts months in advance of the delivery date. Guaranteed display advertising is a multi-billion dollar industry and thus, intelligent pricing of guaranteed contracts has a direct impact on publishers' revenue.

The problem of pricing guaranteed contracts, however, has a few key differentiating factors that require the development of new pricing methods. First, as mentioned above, guaranteed contracts are sold months in advance of the delivery date, and at various points in time during this period. For instance, it is not uncommon to sell some guaranteed contracts a year in advance to some advertisers, a few months in advance to some other advertisers, and just a few days in advance to other advertisers. Second, each user visit can be described by hundreds of attribute values, and advertisers can potentially target (and hence require the publisher to price) any of the many trillions of combinations of these attribute values. For instance, each user visit is typically characterized by the demographics of the user (age, gender, location, etc.), explicitly stated interests of the user (e.g., travel, sports), implicitly inferred interests of the user (e.g., planning a vacation), characteristics of the Web page being visited (e.g., Sports page, Travel page), characteristics of the system being used by the user (e.g., PC vs. mobile, dial-up vs. broadband, IP address location), and so on. Given these attributes, different advertisers may target different combinations; for instance, one advertiser may target Males in California who visit Sports

9*Work done while at Yahoo! Inc

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$10.00.

pages, while another advertiser may target users between 20 and 30 years of age who are planning a vacation.

Due to the above factors, existing pricing methods are not directly applicable to the problem of pricing guaranteed contracts. For instance, let us consider traditional yield optimization methods used in the context of industries such as airlines and retail. Such methods typically model the self- and cross- elasticities of demand at various price points, and try to optimize for the product prices that maximizes revenue [11]. However, these methods are only viable for a relatively small number of products (say, 100s of products), and are not applicable for the trillions of overlapping products as in display advertising, where two contracts rarely have the same targeting combination, and computing cross-elasticities for such a large number of products is impractical at best. As another example, let us consider one of the primary pricing methods used today in online advertising: real-time auctions, which are extensively used in sponsored search [6] and non-guaranteed display advertising [9]. A related pricing method is combinatorial auctions [5], whereby different buyers can specify different combinations of inventory of interest (similar to different targeting combinations), and the problem is to find the yield-maximizing way to allocate the inventory. Since guaranteed contracts are sold well in advance, and at different points in time, they are not amenable to auctions because not all advertisers buy inventory during the same period of time (in this sense, it is analogous to airline ticket pricing, where not all passengers can be forced to buy tickets during the same period of time). In order to conduct an auction where the bidders arrive at various times, we would have to set appropriate reserve prices; for which we would again have to model elasticities for an impractically large number of overlapping products (see also Section 7).

Finally, another key differentiating aspect of the guaranteed display advertising market is the following: due to the sheer number and complexity of products offered, the terms of guaranteed contracts are most commonly decided through manual negotiations between a sales person representing the publisher and a media-buyer representing the advertiser. The sales person is responsible for offering a combination of the available products that best fits the advertisers needs while at the same time extracting the maximum possible value for the sold user visits from the advertiser (e.g., by marking up the price or offering discounts to close a sale, as appropriate). Thus, the price generated by any automated pricing system is only used by the sales person as a starting point in the negotiations and is consequently non-binding. However, we also note that since each sales person constantly books contracts with "new" targeting combinations that have never been booked before, we cannot simply rely on the sales person to negotiate a price without any guidance from a pricing system, because the sales person needs some guidance as to the starting point of the negotiation. Accordingly, we consider the objective of an automated pricing system for guaranteed contracts to be one that provides the sales person information regarding the market value of the user visits that the advertiser is interested in purchasing, which can then be used as a good starting point for negotiations.

1.1 Contributions

To address the above issues, we propose a new method for pricing guaranteed contracts. This method is based on two key ideas. The first idea is to price each contract based on the *value of the individual user visits* that are expected to be delivered to the contract. Using individual user visits as the basic unit of pricing has several advantages over looking only at contracts. First, it helps solve the problem of dealing with the combinatorial explosion of overlapping products; by mapping each product to (a sample of) the set of user visits that are eligible for the product, we can better model and understand the overlap and intersection between the various products. Second, it helps solve the sparsity problem, i.e., the problem of pricing contracts that have very few or no other contracts with identical targeting combinations. As an example, consider a contract that targets Computer Scientists living in Raleigh. Using standard techniques, we could price this contract based on the attributes "Computer Scientist" and "Raleigh," but unfortunately, even with years of contracts, the data is likely to be too sparse to be useful. (How many contracts can we expect to see that target either of these attributes?) However, by mapping this contract to user visits, which may be eligible for many other contracts with different targeting attributes, we can solve the sparsity problem. Finally, the idea of pricing a contract based on the value of the user visits also provides more transparency in pricing: each advertiser pays for what he/she expects to get.

The second key idea behind the proposed method is to leverage the implicit pricing feedback and correction embedded in the current guaranteed sales process. As we mentioned earlier, guaranteed contracts are sold through negotiations between sales persons and advertisers that results in the final price for the contract being higher or lower than the price quoted by the pricing engine. Our key observation is that the final negotiated price implicitly encodes some information about the market conditions, willingness of advertisers to pay a given price, etc. Consequently, we can use the final negotiated prices for recent historical contracts to infer the value of user visits.

Of course, the main open question that remains is how to price individual user visits based on historical contracts. We propose and evaluate two alternative techniques to solve this problem — WAP and MIN-VAR. WAP, or Weighted Average Pricing, postulates that the value of a user visit should be as close as possible to the final negotiated price of *each* eligible historical contract (normalized by the proportion taken up by each contract). If we were to mathematically solve for this objective, it turns out that the value of each user visit is the weighted average of the prices (per user visit) of eligible historical contracts (hence the name WAP). For example, suppose there was a user visit corresponding to a Computer Scientist living in Raleigh visiting a Sports page. If there were two historical contracts interested in the user visit, but 60% of such user visits were given to the first contract (priced at \$1 CPM - Cost per "Mille" or 1000 user visits) while 40% of such user visits were given to the second contract (priced at \$4 CPM), then WAP would price the user visit as the weighted average price of the contracts. In this case, the price would be $60\% \times 1 + 40\% \times 4 = \$2.2$ CPM.

As mentioned above, the second technique for pricing user visits is called MIN-VAR. MIN-VAR postulates that the *sum* of the values of each eligible user visit should approximately be equal to the total cost of the contract (notice the subtle difference between this postulate and the WAP postulate). Essentially, MIN-VAR treats the total cost of each historical contract as a soft constraint, and tries to find user visit values that when added up, correspond to the total cost, and hence satisfy the constraints to the maximum extent possible. Since there may be many different assignment of prices to user visits that can result in the best possible satisfaction of the constraints, MIN-VAR tries to minimize the variance

(hence the name MIN-VAR) between the prices of user visits eligible for the same contract, in the absence of any information to the contrary. MIN-VAR thus ensures that we make as few assumptions as possible on the user visit prices given the historical contract prices.

We have implemented the WAP and MIN-VAR algorithms, and evaluated them using real historical contracts and prices. Specifically, we use a set of historical contracts before time t to train the WAP and MIN-VAR models, use them to price historical contracts after time t, and calculate their fit to the final negotiated booked price (the historical prices were computed using a different pricing method than WAP and MIN-VAR, hence the results are not biased towards WAP or MIN-VAR). Our results show that WAP generally outperforms MIN-VAR because WAP is more robust to variations in booked prices. In fact, somewhat surprisingly, WAP actually has a better fit to the final negotiated booked price than the original pricing method, even though the original pricing method was used by the sales team as a starting point for negotiating the final price.

We note that there are two other aspects that are related to the guaranteed pricing problem considered here, but are beyond the scope of this paper. The first aspect is forecasting a sample of the set of user visits that are eligible for a contract that may be booked months in advance. However, this problem has to be (and in reality is) solved in any case for a guaranteed system because a publisher needs a fairly accurate forecast in order to guarantee a certain number of user visits to contracts and avoid penalties. The second aspect is correcting prices for current market conditions, specifically those conditions that are not captured by prices of recently booked contracts. For instance, prices may need to be dropped (increased) at the start (end) of a recession due to a sudden drop (increase) in demand not reflected in recent historical prices. While there are techniques to address such price corrections based on the amount of inventory sold, etc., these are beyond the scope of this paper.

1.2 Summary and Roadmap

In summary, the main contributions of this paper are:

- A model and formal statement of the guaranteed pricing problem (Section 2).
- The WAP algorithm for pricing guaranteed contracts (Section 3).
- The MIN-VAR algorithm for pricing guaranteed contracts (Section 4).
- An evaluation of the WAP and MIN-VAR algorithms using historical guaranteed contracts and prices (Section 5).

2. MODEL AND PROBLEM STATEMENT

We first describe the supply and demand model, before formally stating the pricing problem.

2.1 Supply and Demand Model

The basic unit of supply is an individual user visit, which is identified by a set of attribute-value pairs that include information about the user and the context of the visit. Specifically, a user-visit may be defined by

- **User Information** : demographic information such as age, gender and income bucket; inferred behavioral attributes such as "*interest in sports*" or "*shopping for car*"; geographic information such as country, state, DMA, city or zip.
- **Content Information** : Information regarding the specific web page visited in the publisher's content hierarchy such as site or section; specific keywords related to the visited webpage.
- **Time Stamp** : The time stamp of the user visit in UTC time.

Let us suppose that there are $k = 1, \ldots, K$ attributes that specify the user and content information, with the set of allowable values for attribute k being denoted by $\mathcal{A}_k$. Thus the universe of possible attribute combinations is $\mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_K$.

The basic unit of demand is a guaranteed contract. A guaranteed contract is essentially a guarantee that a publisher provides to show an ad from a particular advertiser to a set of users whose attribute-values fall in a subset that is desirable to that advertiser. In particular, a typical guaranteed-delivery contract (denoted c) may specify the following:

- **Targeting Predicate** : The targeting predicate is a Boolean expression over the attribute space $\mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_K$ that specifies the set of user visits eligible for the contract. For example, the targeting predicate of a guaranteed contract that targets males in the US visiting non-Spanish pages with content topics NBA or NFL or females not in the US would be represented as:

  $(\texttt{Gender} \in \{\texttt{Male}\} \wedge \texttt{Country} \in \{\texttt{US}\} \wedge \texttt{Language} \notin \{\texttt{Spanish}\} \wedge \texttt{Content-Topic} \in \{\texttt{NBA}, \texttt{NFL}\})$.

  It is easily seen that the targeting predicate could specify any subset of the universe of attribute-values of a user-visit, i.e., an element of the set $2^{\mathcal{A}_1 \times \ldots \times \mathcal{A}_K}$.
- **Flight Duration** : The flight duration specifies the start and end times of the query, in UTC time. For instance, the start time of a query could be 24 May 2010 at 10am and the end time of the query could be 14 Aug 2010 at 11pm.
- **Impression Goal** : The number of user visits for which the advertiser's creative needs to be displayed.

2.2 Problem Statement

Given the above supply and demand models, the problem of pricing guaranteed contracts can be stated as follows. The goal is to find an appropriate pricing function Q that maps any allowable combination of targeting predicate, duration and impression goal to a corresponding price, i.e.,

$$Q : 2^{\mathcal{A}_1 \times \ldots \times \mathcal{A}_K} \times [t, t+T] \times [t, t+T] \times \mathbb{R}^+ \rightarrow \mathbb{R}^+ \quad (1)$$

Stated in this fashion, the scale of the pricing problem becomes evident. A large publisher commonly has many Web pages and offers hundreds of different user attributes that advertisers can target. Consequently, the input to the pricing function can be many trillions of pricing combinations. Note that even though advertisers may not purchase all such combinations, it is not known a priori which subset of the combinations they are interested in; consequently, the pricing function should be able to dynamically price any one of the combinations.

Pricing guaranteed contracts thus presents a formidable challenge. In the next section, we present an overview of our solution approach to attack this problem.

3. SOLUTION OVERVIEW

On the face of it, the guaranteed pricing problem may look similar to the pricing problems faced in other industries that sell multiple products, such as airlines, hotels, and retail. Indeed many of these yield management problems have been very well studied for over 30 years [11], and they typically work by estimating the demand curve for a single product (self-elasticity), as well as the interactions between the demand curves of multiple products (cross-elasticity). However, directly applying these techniques to the guaranteed pricing problem is impractical for two reasons.

First, the sheer number of products is many orders of magnitudes higher than that considered in other domains, and consequently, there is extreme sparsity in the data to estimate self-elasticities (for instance, it is very rare for two advertisers to buy exactly the same targeting combinations, let alone durations). Furthermore, since there are manytrillions of products, estimating cross-elasticities at this scale is impractical, again due to data sparsity.

The second reason that existing techniques do not directly apply is more subtle: even though the different targeting combinations can be viewed as different products, they may actually be competing for the same set of user visits. For instance, targeting users whose income is above $100,000 a year in the US may be almost the same as targeting users in the large cities of the US. Understanding and capturing correlations and cross-elasticities between a large number of such overlapping products soon becomes impractical.

In order to address the above issues, we propose a new and alternative way to price guaranteed contracts, as described below.

3.1 Pricing Based on Expected User Visits

The key idea is to view the price of the contract as the sum of the values of the individual user visits that the contract is expected to get. More formally:

The total cost of a contract is the sum of the values of the expected user visits that will be delivered to that contract. In particular, suppose that a contract c will be delivered user visits $\mathcal{I}_c$, and the price of each user visit $i \in \mathcal{I}_c$ is p_i, then the total cost Q_c of the contract c is given by,

$$Q_c = \sum_{i \in \mathcal{I}_c} p_i \tag{2}$$

The intuition behind this idea is the following. Although an advertiser pays a single price for a guaranteed contract, not all of the user visits that the advertiser obtains have the same economic value. For instance, an advertiser who targets users visits to Finance Web pages may get some user visits corresponding to users from high-income New York City zip codes, and some impressions from poor rural areas. Even though the advertiser pays for the entire "package," the economic value of these impressions for advertisers is likely to be very different. Thus, from the advertiser's point of view a guaranteed contract is a bundle of heterogeneous objects (a mix of more and less valuable user visits), and hence the the price of a guaranteed contract should reflect the overall value of this mix.

Pricing based on expected user visits enables us to address some of the problems that we encountered with traditional yield optimization techniques. For instance, dealing with trillions of seemingly unrelated products that nevertheless overlap can be handled effectively by mapping each product to their corresponding user visits, and using this overlap to guide pricing. For instance, it is possible to determine that the set of eligible user visits for a contract targeting users whose income is above $100,000 a year in the US is almost identical to the set of user visits eligible for a contract targeting users in the large cities of the US.

The above idea also enables us to address some of the scalability problems of the traditional yield optimization techniques when dealing with such a large number of products. Although there are many billions of individual user visits, it is sufficient for the purpose of pricing to work with a *sample* of user visits for each contract. As we shall experimentally evaluate in Section 6, a sample of a thousand user visits per contract is often sufficient to calculate prices with high-fidelity, while also capturing the effects of overlap. In a sense, this approach may almost seem paradoxical: there are billions of user visits, but only a few hundreds of thousands of booked guaranteed contracts. Yet, we use the prices of individual user visits to price contracts. The reason we do so, of course, is that the number of *possible* contracts is of the order of many trillions, and we can work with a small sample of user visits to obtain the prices.

Of course, there is still the open issue of how to determine the prices of individual user visits, which we address next.

3.2 Pricing User Visits based on Historical Contracts

In order to price user visits, we leverage the prices of recent historical contracts, and some of the feedback and market correction mechanisms inherent in the current guaranteed contract sales process. Specifically, current guaranteed contracts are typically not sold using a self-serve interface, but are sold through sales agents interacting either directly with advertisers, or with agencies working on behalf of advertisers. Consequently, the price produced by the pricing engine is only used as a starting point for negotiations, and can be adjusted upwards or downwards depending on market dynamics, competition, advertisers' willingness to pay, and so on. What this implies, however, is that the final negotiated prices of historical contracts is a summary of various exogenous factors that have an impact on prices, and thus our proposal is to use these to price individual user visits.

More formally, the goal is to find a pricing function P that takes in a user visit i , and a set of matching historical contracts $\mathcal{H}_i$ along with their final negotiated prices and other metadata, and outputs the price of the user visit:

$$P : i, \mathcal{H}_i \rightarrow \mathbb{R}^+ \tag{3}$$

In the subsequent sections of this paper, we propose two implementations of this pricing function — WAP and MIN-VAR.

At this point, it is important to note that the proposed approach does not aim to come up with a provably revenue optimal final price for a contract. Rather, the goal is to produce a contract price (by producing the appropriate user visit prices) that provides a good starting point for sales negotiations by taking into account recent marketplace corrections that are captured in recently negotiated historical prices.

3.3 Notation

In the following two sections we describe the WAP and MIN-VAR pricing methods for estimating user visit prices

based on historical contract prices. We now introduce some notation that will be used in these sections.

In general, we continue to use i as a generic identifier for a user visit and j as a generic identifier for a historical contract. We denote the set of historical contracts used to price user visits by $\mathcal{H}$ and as before, we denote the subset of $\mathcal{H}$ that matches a user visit i by $\mathcal{H}_i$. The final negotiated price per user visit for a contract $j \in \mathcal{H}$ is denoted q_j and the price of a user visit i generated by WAP or MIN-VAR is denoted by p_i.

We denote a future contract to be priced as c and let $\mathcal{I}_c$ denote the sampled set of user visits that will be delivered to c. Since the user visits in $\mathcal{I}_c$ have time stamps in the future, in practice we would have to translate these time stamps to the past to find a set of matching historical contracts. Here, for simplicity of notation, we let $\mathcal{H}_i$ denote the set of historical contracts matching a future user visit $i \in \mathcal{I}_c$ with the implicit understanding that the user visit i has been translated back in time to find the matching historical contracts.

4. WAP PRICING

The main idea behind WAP pricing is the following: *the price of each user visit is as close as possible to the final negotiated price of an eligible historical contract (normalized by the number of impressions)*. Of course, a user visit may have multiple eligible historical contracts, and in this case, the goal is to make the deviation as small as possible across all these contracts. These aspects are formalized next.

4.1 Formal Model

The WAP price for a user visit is obtained by minimizing a weighted least squares objective function, where the prices of historical contracts are the independent variables and the price of the user visit as the dependent variable. That is, given a user visit i and a set of eligible historical contracts $\mathcal{H}_i \subset \mathcal{H}$, the price p_i is obtained as:

$$p_i \in \arg\min_{p \geq 0} \left\{ \sum_{j \in \mathcal{H}_i} x_{ij} \left(q_j - p\right)^2 \right\} \tag{4}$$

Here, $x_{ij} \geq 0$ is a weight that captures the importance of contract j in determining the price of i, and is used to capture the fact that not all historical contracts have the same influence on the price of a user visit. For instance, a narrowly targeted eligible contract that targets a small number of user visits is likely to have a larger impact on the user visit price than a contract that targets potentially billions of user visits, some like i and some very different user visits. One way of capturing this weight systematically is to view it as an ad serving probability, i.e., the probability that the ad server will serve a user visit such as i to a contract j. We consider an approximation to this interpretation in the experimental section.

We now turn to the issue of efficiently solving for the user visit prices that meet the above objective.

4.2 Solving for User Visit Prices

Since the objective function to be minimized on the right hand side of (4) is a simple quadratic function of p, the price p_i can be written in closed form as

$$\begin{aligned} p_i &= \max\left\{ \frac{\sum_{j \in \mathcal{H}_i} x_{ij} q_j}{\sum_{j \in \mathcal{H}_i} x_{ij}},\ 0 \right\} \\ &= \frac{\sum_{j \in \mathcal{H}_i} x_{ij} q_j}{\sum_{j \in \mathcal{H}_i} x_{ij}} \end{aligned} \tag{5}$$

since $q_j, x_{ij} \geq 0$ for each $j \in \mathcal{H}_i$. Thus, the user visit price p_i can be simply computed using the above formula. Having calculated the p_i for all $i \in \mathcal{I}_c$, the price per user visit for contract c is calculated using (2).

5. MIN-VAR PRICING

The MIN-VAR algorithm adopts a subtly different approach than WAP, and its main idea is the following: *the price of the sum of eligible user visits is as close as possible to the final negotiated price of a historical contract*. Accordingly, the MIN-VAR algorithm solves a linear regression model to infer the prices of individual user visits from the prices of the historical contracts. We now describe the model more formally, before describing how it can be solved efficiently.

5.1 Formal Model

Let us assume that for each contract $j \in \mathcal{H}$, we have a sampled set of user visits eligible to be served to j and denote this set of user visits as $\mathcal{I}^H$. For each such user visit $i \in \mathcal{I}^H$, we let μ_i denote the "weight" of the user visit i in the sample. We can think of μ_i as the number of user visits that are represented by the single user visit i in the sample. As in the case of WAP, we let x_{ij} denote the ad server probability that the user visit i will be served to a contract j. Now, we can define the total number of user visits that will be served to contract j as

$$D_j := \sum_{i : j \in \mathcal{H}_i} \mu_i x_{ij}$$

Then, the MIN-VAR algorithm solves the following optimization problem to determine the price p_i of a user visit i (simultaneously finding error terms z_j).

$$\begin{aligned} \min \quad & \sum_{j \in \mathcal{H}} \left\{ \sum_{i : j \in \mathcal{H}_i} \mu_i x_{ij} \left(p_i - q_j\right)^2 + W D_j z_j^2 \right\} \\ \text{s.t.} \quad & \sum_{i : j \in \mathcal{H}_i} \mu_i x_{ij} p_i = \left(q_j - z_j\right) D_j \\ & p_i \geq 0 \qquad \forall\, i \in \mathcal{I}^H \end{aligned} \tag{6}$$

It is easy to see that the objective function is the weighted sum (with W being the relative weight) of two different functions. If we ignore the first term in the objective involving the user visit prices p_i, we see that what remains is a traditional least squares optimization problem that tries to fit the user visit prices to add up as close as possible to the negotiated contract prices. However, even though we sample only a thousand user visit per contract and there are only a few thousand contracts in $\mathcal{H}$, the total number of user visits in $\mathcal{I}^H$ is in the millions, which is a few orders of magnitude higher than the number of historical contracts. Thus, there would be multiple sets of user visit prices that would minimize the error in fit to the contract prices.

In order to choose between these optimal solutions we use an approach similar to entropy maximization. In particular, we wish to choose a solution where the variation in the prices

of all the user visits that served a particular contract to be as small as possible (hence the name MIN-VAR). Stated differently, unless there is specific information forcing the prices of any two user visits to be different we would like their prices to be equal. We achieve this objective by adding the term

$$\sum_{i:j\in\mathcal{H}_i} \mu_i x_{ij} \left(p_i - q_j\right)^2$$

for each contract j, which is essentially the variance (relative to the contract price) of the individual user visit prices used to serve the contract.

5.2 Relationship to WAP Pricing

Before we go on to describe how to efficiently compute user visit prices for MIN-VAR, we would like to point out the relationship between the WAP and MIN-VAR algorithms. Consider for a moment the objective function in (6) with $W = 0$. This can be written as

$$\min \sum_{j\in\mathcal{H}} \sum_{i:j\in\mathcal{H}_i} \mu_i x_{ij} \left(p_i - q_j\right)^2 \;=\; \min \sum_{i\in\mathcal{I}^H} \mu_i \left\{ \sum_{j\in\mathcal{H}_i} x_{ij} \left(p_i - q_j\right)^2 \right\} \quad (7)$$

It is easily seen that for each user visit i, the MIN-VAR objective is exactly the same as the WAP objective. Since for $W = 0$ the regression portion of the MIN-VAR objective (and hence the constraints) do not play any part, the user visit prices generated by the WAP and MIN-VAR algorithms are identical.

5.3 Solving for User Visit Prices

We now turn to the problem of efficiently computing user visit prices that satisfy the MIN-VAR objective. At first glance, a simple solution would be to solve the problem in two stages, i.e., first solve the regression problem and then find the minimum variance solution among the set of optimal solutions to the regression problem. However, we note that underdetermined regression solutions are highly susceptible to outliers in the data — a contract with an anomalous price could skew the prices of all its eligible user visits. To avoid this problem, we solve for both objectives together with a relative weight W that is tuned based on empirical experiments so as to be resistant to outliers, while at the same time trying to fit user visit prices to the historical contract prices.

Another issue with solving the above objective is that, unlike WAP, the price of a user visit is related to the price of *other* user visits eligible for the historical contracts, including those user visits that may not even be eligible for the new contract being priced. Consequently, the problem needs to be solved for a large set of user visits only indirectly related to the contract being priced, and it is typically impractical to solve this problem online to produce a price for the sales agent. Solving and storing the user visit prices offline also has its challenges because we cannot possibly compute user visit prices for *every possible* sample that might be requested for the trillions of products. So, we are thus faced with a dilemma: computing the prices online is too expensive, but computing the prices offline for every possible user visit that could be sampled for a query is impractical. What do we do?

Our proposed solution works in two steps. In the first step, we solve the optimization problem (6) offline with a sample of user visits relevant to the historical contracts (and not the new contract to be priced). Based on this solution, we store the optimal dual values [3] (denoted by β_j^*) per contract corresponding to each equality constraint in (6). In the second (online) step, given a set of user visits expected to be served to the contract to be priced, we use the stored β_j^* values for the historical contracts to rapidly compute the price for each new user visit. As we shall now show, this two step approach pushes most of the heavy-weight computation offline, and still enables us to compute the user visit prices rapidly online.

We now describe how we can quickly compute the user visit prices online using the dual values of the optimal solution (the β_j^* values). From the duality theory for convex optimization problems [3], it can be shown that given an optimal vector $\beta^* := (\beta_j^* : j \in \mathcal{H})$ of Lagrange multipliers for the equality constraints of (6), any vector $p := (p_i \;:\; i \in \mathcal{I}^H)$ such that $p \geq 0$ and vector $z := (z_j \;:\; i \in \mathcal{H})$ is an optimal set of user visit prices and slacks for (6) if and only if the following complimentary slackness condition holds

$$(p, z) \;\in\; \arg\min_{p\geq 0} L(p, z, \beta^*) \quad (8)$$

where for any p, z and β the Lagrangian function $L(p, \beta)$ is defined as

$$L(p, z, \beta) \;:=\; \sum_{j\in\mathcal{H}} \left\{ \sum_{i:j\in\mathcal{H}_i} \mu_i x_{ij} \left(p_i - q_j\right)^2 + W D_j z_j^2 \right\} + - \sum_{j\in\mathcal{H}} \beta_j \left\{ \sum_{i:j\in\mathcal{H}_i} \mu_j x_{ij} p_i \;+\; D_j z_j \;-\; D_j q_j \right\} \quad (9)$$

Rearranging the terms of (9) and using (8), we get that

$$\min_{p\geq 0} L(p, z, \beta^*) \;=\; \sum_{i\in\mathcal{I}^H} \mu_i \min_{p_i\geq 0} \sum_{j\in\mathcal{H}_i} \left\{ x_{ij} \left(p_i - q_j\right)^2 - \beta_j^* x_{ij} p_i \right\} + \sum_{j\in\mathcal{H}} \min_{z_j} \left\{ D_j \left(W z_j^2 \;-\; \beta_j^* z_j \right) \right\} \;+\; \beta_j^* D_j q_j \quad (10)$$

It is easy to see that (10) can be solved in closed form individually for each p_i and z_j. Thus, from (10), we see that the optimal user visit prices p_i and the slacks z_j can be obtained from the optimal β_j^* values using

$$p_i \;:=\; \max\left\{ 0, \frac{\sum_{j\in\mathcal{H}_i} \left(q_j + \frac{\beta_j^*}{2} \right) x_{ij}}{\sum_{j\in\mathcal{H}_i} x_{ij}} \right\} \quad (11)$$

$$z_j \;:=\; \frac{\beta_j^*}{2W} \quad (12)$$

Thus, the above equation is used to compute the user visit prices online. Once the user visit prices are computed, the contract price is then computed using (2).

6. EXPERIMENTAL EVALUATION

We have implemented the WAP and MIN-VAR algorithms, and benchmarked them against the algorithm currently used to recommend prices to sales persons for a large Web publisher. For the experiments, we used historical guaranteed contracts from the publisher, along with their actual negotiated prices.

We begin by motivating the metrics and the design we chose for our evaluation. We then present the details of the system architecture used to implement the tests, and finally present our results.

6.1 Experiment Design and Metrics

Since WAP and MIN-VAR (like the algorithm currently in use) generate prices that are used as a starting point for negotiations by sales persons, we evaluate the algorithms by measuring the fit of the generated prices to the actual negotiated prices. This evaluation methodology is similar to that used in [8] for evaluating price recommendations in a related but different context (please see Section 7 for more details).

First, we choose a set of contracts that were booked in some time interval in the past say $[t_s^H, t_e^H]$. Note that we can retrieve both the final negotiated price and the price recommended by the current pricing algorithm for each contract in our test set. We use another set of historical contracts that were active before t_s^H to price our test contracts using WAP and MIN-VAR. Then, we compare the quality of the prices recommended by each algorithm (WAP, MIN-VAR and the current algorithm) by measuring the fit against the final negotiated prices of our test contracts. To be specific, if $\{q_j : j = 1, \ldots, K\}$ is the set of negotiated prices for our test contracts and $\{\hat{q}_j, j = 1, \ldots, K\}$ is the set of prices generated by one of the candidate algorithms, we use the following standard metrics to evaluate the fit.

$$R^2 := 1 - \left\{ \frac{\sum_{j=1}^{K} (q_j - \hat{q}_j)^2}{\sum_{j=1}^{K} \left(q_j - \left(\frac{\sum_{i=1}^{K} q_i}{K} \right) \right)^2} \right\}$$

$$MAPE := \left(\frac{1}{K} \right) \sum_{j=1}^{k} \frac{|q_j - \hat{q}_j|}{|q_j|}$$

(MAPE refers to Mean Absolute Percentage Error).

Based on our stated objective to provide a good starting price for negotiations, an ideal pricing algorithm would be one that was able to somehow predict exactly what price the advertiser would be willing to pay and display this as the starting price. In this case, the final negotiated contract price would be equal the system suggested price. Consequently, the R^2 metric for an experiment conducted as above would be 1 and the MAPE metric would be 0. Thus, in our experiments, R^2 and MAPE metrics are intended to measure the accuracy of the WAP and MIN-VAR generated prices as predictors of the final negotiated price of the corresponding contracts.

It is very important to note that the aforementioned test is inherently *biased in favor of* the algorithm currently used for price recommendations. This bias arises because the price recommended by the current algorithm was in fact, the basis for the negotiations that ended in the final booking price for each test contract. In the extreme case for example, if sales persons performed no negotiations and each advertiser booked contracts at the price recommended by the pricing algorithm, the R^2 value for the current pricing algorithm would be 1 (a perfect fit), even though this is not a true reflection of the extent of the fit. Fortunately, in our case, the sales persons do perform negotiations, so the bias is somewhat limited but it is nevertheless present due to strong correlations that we observed between the overall monthly average booking price and the overall monthly average price recommended by the current algorithm.

We note that it might be possible to address some of the bias issues by performing a live online test, whereby a subset of the sales persons use our proposed algorithms in the actual bookings. While such a test is in our plans for future work, it would require significant online experiment design to address the errors inherent in partitioning a relatively small number of sales persons; further, it would have to be carried out over a long period of time (many months or even a year) given the lead time for booking guaranteed contracts.

6.2 System Architecture

6.2.1 WAP and MIN-VAR algorithms

Figure 1 shows the system architecture that implements the proposed algorithms.

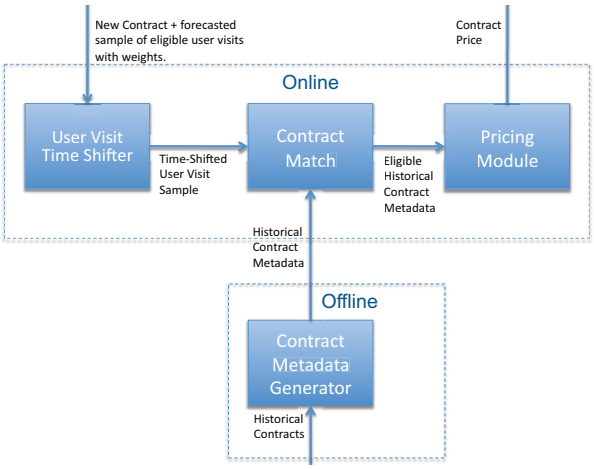

Figure 1: Algorithmic Architecture Diagram

For each test contract to be priced, the pricing system receives a sample of forecast user visits $\mathcal{I}_c$. The sample of forecast user visits is obtained from a supply forecasting system used for booking guaranteed contracts.

Both the WAP and MIN-VAR algorithms are implemented with an offline and an online component.

- Offline, the Contract Metadata Generator chooses an appropriate set of recent historical contracts $j \in \mathcal{H}$ and populates Contract Match (which is essentially a database for the historical contracts) with the final negotiated price q_j, the quantity $\alpha_j = (D_j/S_j)$ where D_j denotes the set of number of user visits delivered to contract j and S_j denotes the total number of user visits *eligible* for contract j. This ratio α_j is used in our experiments as a heuristic approximation for the probability x_{ij} that any user visit will served to contract j. The intuition is that if the total number of user visits demanded by a contract is very close to the total number of eligible user visits, the any particular eligible user visit does indeed have a high probability of being delivered to that contract. For the MIN-VAR algorithm, the Contract Metadata Generator also stores the optimal dual values β_j^* from solving (6) in Contract Match.

- When the set of user visits $\mathcal{I}_c$ is sent for pricing, the User Visit time-shifter first translates the time stamps of each user visit back in time to the period when the contracts in $\mathcal{H}$ were active. Then, for each such user visit, Contract Match returns a set of eligible historical contracts along with the q_j, α_j and if applicable the β_j^* values. Then, the Pricing Module uses either (5) (for WAP) or (11) (for MIN-VAR) setting $x_{ij} = \alpha_j$ to find the price of each user visit. Then, the price of the test contract is calculated as $\sum_{i \in \mathcal{I}_c} p_i$.

6.2.2 The Current Pricing Algorithm

Since we are using the current pricing algorithm (henceforth, denoted as CURRENT) as a benchmark for the predictive performance of our proposed algorithms, we provide a brief description of this algorithm in this section. The CURRENT algorithm sets prices by finding demand elasticities using historical data and then solving a revenue maximization problem for the prices. Of course, the CURRENT algorithm faces the same problems arising from the sparsity of data relative to the number of different but overlapping products offered. However, it overcomes this difficulty by aggregating all products into a few buckets and treating all products within the same bucket as one. Given a contract to be priced, CURRENT first finds a "base price" for the contract as the price of the bucket that the corresponding product would fall in. Then based on each extra attribute set specified in the contract's targeting predicate, CURRENT adds on a percentage markup to the base price to finally arrive at the price of the contract.

6.3 Experiment 1

First, in order to test the relative performance of both WAP and MIN-VAR, we chose a set of guaranteed display contracts that had been booked within the span of one month against one particular ad position on the chosen publisher's content network. For our historical contract set, we chose contracts booked against the same ad position that had been active in the three months prior to the booking month of our test contract set. We priced our set of test contracts using the WAP algorithm and the MIN-VAR algorithm with 5 different values of the weight parameter W. Table 1 shows the R^2 and MAPE (normalized as a percentage difference from the MAPE value for the CURRENT algorithm) metrics calculated on the prices generated by each of our candidate algorithms.

Algorithm	R^2	MAPE
CURRENT	0.5396	-
WAP	0.5451	-1.9%
MIN-VAR ($W = 1$)	0.5192	+2.1%
MIN-VAR ($W = 10$)	0.4911	+31.7%
MIN-VAR ($W = 100$)	0.3188	+69.7%
MIN-VAR ($W = 1000$)	0.0076	+128.7%
MIN-VAR ($W = 10000$)	0.0003	+402.2%

Table 1: Experiment 1 : R^2 and MAPE

Immediately, we note the surprising fact that the WAP algorithm shows both a higher R^2 and lower MAPE metric compared to the CURRENT algorithm despite the inherent bias towards CURRENT that we described in Section 6.1. This result definitely reinforces our confidence in the user visit based approach to pricing guaranteed contracts.

Also, it is clear from both the MAPE and R^2 values in Table 1 that WAP is able to better predict the prices of the contracts in the test set. Also, recalling that WAP produces prices identical to MIN-VAR when $W = 0$, we note that the performance of the MIN-VAR algorithm steadily decreases as with increasing W. The reason for this becomes clear when we note the connection between the weight W, the variables z_j in the MIN-VAR model and the impression prices generated by the model. From (11) and (12), we can write the prices generated by MIN-VAR as:

$$p_i = \max\left\{0, \frac{\sum_{j\in\mathcal{H}_i}(q_j + Wz_j)\,x_{ij}}{\sum_{j\in\mathcal{H}_i} x_{ij}}\right\} \quad (13)$$

Thus, the user visit prices depend directly on the fit of the regression, i.e., whether each historical contract price can be well represented as the sum of its user visit prices. In this situation, especially for a large value of W it is easy to see that even the presence of a single contract with a price that is an outlier (compared to the prices of other similar contracts) can lead to dramatically different user visit prices. We believe that it is the presence of such contracts with anomalous prices in our historical set that causes the prediction performance of the MIN-VAR algorithm to degrade as the weight W increases.

6.4 Experiment 2

Based on the results we observed in our initial set of experiments, we decided to perform a more extensive set of tests using only the WAP algorithm. For this test, we chose a much larger set of guaranteed delivery contracts booked over the span of 12 months on multiple properties in our publisher's content portfolio (Note that the snapshot of results we show is deliberately incomplete and subject to some anonymization and thus does not necessarily reflect the publisher's portfolio at any particular time). For each month of test contracts, we used contracts that were active in the previous three months as the historical set.

Having generated a price for each contract in the test set using the WAP algorithm for all our test contracts, we compared the R^2 metric for WAP and CURRENT at multiple levels of aggregation. In Table 2, we report the overall R^2 value (for all properties together) and the normalized R^2 for WAP as a percentage difference from that for CURRENT for selected properties.

Property	CURRENT	WAP
Overall	0.5826	0.5643
A	-	+81.8%
B	-	-50.2%
C	-	-35.5%
D	-	+222%
E	-	+330.5%
F	-	-42.1%

Table 2: Experiment 2 : Overall performance of WAP vs. current algorithm

Again, we see that even over a large data set of test contracts booked over a year, WAP performs remarkably well given the bias towards CURRENT. However, it is also clear that the percentage difference of the R^2 values of WAP from the R^2 values for CURRENT are seen to vary quite dramatically between specific properties. When we investigated this, we saw that WAP tended to have a better fit to the negotiated price on properties where highly targeted contracts were booked (properties A, D and E). Specifically, for highly targeted contracts that had higher negotiated prices relative to the average price for that property, we found that WAP was able to recognize the true value of the user visits that would be delivered to the contract and predict a more accurate price for it than CURRENT. Figure 2 shows an example of a typical property where we noticed this. In this figure, which is best viewed electronically or using a color print-out, we have

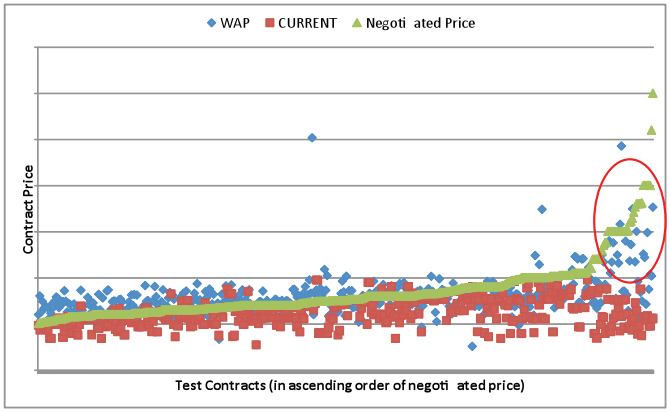

Figure 2: Individual test contract prices where R^2 of WAP is greater than R^2 of CURRENT

plotted the negotiated price and the WAP and CURRENT prices for each test contract. The test contracts have been arranged from left to right in increasing order of negotiated price. The red boundary indicates the region where the WAP prices are seen to be closer to the negotiated prices than the CURRENT prices.

On the other hand, we noticed that WAP tended to have a relatively poorer fit with respect to the negotiated price for properties where a large number of contracts in the test set had been booked at prices very close to that recommended by CURRENT. However, as mentioned in Section 6.1, this is expected because when there is little negotiations performed by sales persons, the results are significantly biased towards CURRENT. Figure 3 shows an example of a typical property where we noticed this phenomenon. As before, in this figure

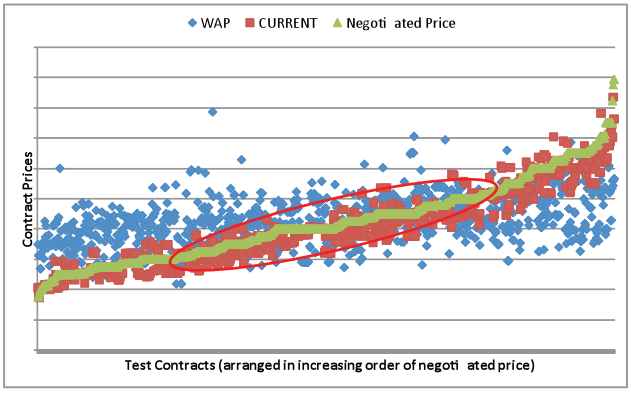

Figure 3: Individual test contract prices where R^2 of WAP is less than R^2 of CURRENT

we have plotted contracts from left to right in increasing order of negotiated price, and the red boundary indicates the test contracts where the negotiated prices is very close to the CURRENT recommended price.

Finally, Figure 4 shows, for a particular property, the monthly R^2 values for WAP normalized to a percentage difference from the corresponding R^2 value for CURRENT; the results are shown for a period of 12 contiguous months (not necessarily from January to December). Clearly, the predictive performance of the WAP algorithm drops off from month 1 to month 7 and then improves again in subsequent months. We were able to correlate this reduction in performance of the WAP algorithm to macro-economic changes in the guaranteed display marketplace that resulted in large changes in the overall demand and negotiated prices. Since the WAP algorithm uses historical contracts to price future contracts, the prices predicted by WAP always lagged the macro-economic changes in the marketplace leading to a reduction in the predictive performance. Accordingly, once the market stabilized, the performance of WAP was seen to improve. We believe this example demonstrates the need for modeling changes in the market that are not reflected in the historical prices. We are currently working on incorporating features that address this need into our models.

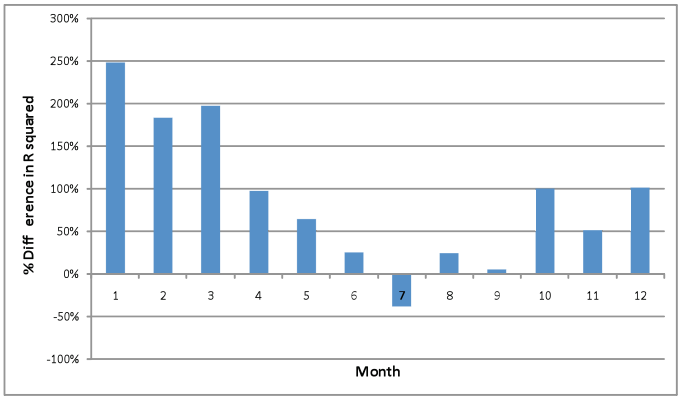

Figure 4: Percentage difference between R^2 values for WAP and CURRENT

6.5 Summary of Experiments

In summary, the experiments show that WAP is overall a better predictor of final negotiated prices than MIN-VAR, mainly because WAP better handles outliers in the negotiated prices. Perhaps most surprisingly, the fit of WAP to the historical negotiated prices often *exceeds* the fit to CURRENT, even though the historical negotiated prices were directly influenced by CURRENT. These results indicate that user visit level pricing is a promising direction to pursue for pricing guaranteed contracts.

7. RELATED WORK

Pricing problems have been extensively studied in the context of many economic environments. Almost any business faces pricing decisions and many concepts of price theory are applicable across a wide range of industries. For example, all monopolists who sell non durable goods are conceptually identical regardless of the industry where they operate. Monopoly profit maximization is a straightforward problem covered in standard economic texts.

The pricing problem becomes very complex when products are differentiated. There is extensive literature that explores the equilibrium strategy of a firm that has to choose price and product characteristics while facing competition from other firms. This problem was first studied in the context of auto industry [2]. A large body of work that followed this paper applied similar techniques to study competition in industries

ranging from consumer electronics [7] to breakfast cereals [10]. The ideas from these articles do not transfer to guaranteed display advertising environment because unlike car makers that choose what products to make and are constrained to offer a relatively small number of products, the product that a publisher has to sell are visits from individual users– thus publishers have a very large set of "products" and no ability to influence product attributes (users are what they are).

The yield optimization problem which considers the question of how a firm should price perishable inventory in the presence of fluctuating demand, has been well studied in the context of many different industries (airline, hotel, car rental, cruise line, retail sales etc.). Talluri and van Ryzin [11] provides an extensive survey of various approaches to tackle this problem. Boyd and Bilegan [4] also discuss the history of automated yield management and its application to E-Commerce. However, most yield optimization approaches used in industries such as airlines or hotels do not scale well to the number of overlapping products typically sold in the guaranteed display market.

Other than for the guaranteed display market, real-time auctions seem to be the pricing method of choice for online advertising. Indeed, sponsored search advertising (based on users searching for particular keywords) is sold exclusively via real-time auctions. We refer the reader to [6] and [12] for a detailed analysis of that market. In fact, there is a relatively small but growing non-guaranteed display advertising market where again user visits are sold via real-time auctions. This has been mainly fueled by the appearance of so-called Ad Exchanges that conduct the real-time auction in exchange for transaction fees. We refer the reader to [1] for more details. Research work related to the design of real-time auctions for non-guaranteed display advertising has only recently started to appear in literature. We refer the reader to [9] for an overview of open research problems in this area.

A combinatorial auction is another mechanism that is appealing when the seller has a large number of goods and buyers show interest in overlapping sets of products. We refer the reader to [5] for a survey of work related to combinatorial auctions. We note that since guaranteed contracts are sold to advertisers at different points in time, for any auction mechanism to be successful, we would need to set appropriate reserve prices. However, in order to set reserve prices that maximize the revenue for the publisher, we would need to estimate the revenue that the publisher could make from the future demand for user visits. In order to do this, we would need to either estimate the self- and cross-elasticities for all possible product combinations (which is impractical) or at least, generate an estimate of the market value of the user visits that the current contract is asking for. Clearly, we would need to use an algorithm like WAP or MIN-VAR in order to do the latter. It is also worth noting that even if we are able to set good reserve prices, solving the allocation problem for such a large number of overlapping products is non-trivial. [13] addresses this issue by grouping user visit attributes to reduce the size of the allocation problem while minimizing the revenue loss due to the grouping.

Finally, [8] considers the problem of providing bid guidance in an online marketplace where advertisers and publishers negotiate on the prices of contracts for offline print media. However, the methods presented in this article do not directly apply to the guaranteed display marketplace for multiple reasons. Most significantly, in this article it is the *market runner* who provides guidance to the *advertisers* on good starting bids in an effort to reduce the information asymmetry present in the market. Thus, the bid guidance here is intended not to maximize the utility for either party, but to make the marketplace more efficient. Further, since the targeting capabilities of online print media are very limited, the product space is many orders of magnitude smaller than that for online media.

8. CONCLUSION

In this paper, we proposed a new approach to pricing guaranteed display advertising contracts, which is pricing based on the user visits that will be delivered to the contract. We proposed two alternative algorithms — WAP and MIN-VAR — based on this approach. We also implemented and evaluated both these algorithms using historical guaranteed contracts booked against the content portfolio a prominent Web content publisher. Our results show that WAP outperforms MIN-VAR due to its increased resilience to outlier prices. Perhaps more surprisingly, our results also show that WAP is often a better predictor of the final negotiated price for historical contracts than even the initial historical price, which was used as a starting point for the negotiations!

As part of future work, we are exploring the use of online bucket tests to derive metrics that are not inherently biased towards the current pricing algorithm. Another promising direction for future work is to investigate the use of signals besides user visits, such as seasonality and macro-economic factors, which might influence the price of a contract.

9. REFERENCES

[1] Right media : `http://www.rightmedia.com/`. More info at `http://www.rightmedia.com/right- media- 101/`.

[2] S. Berry, J. Leveison, and A. Pakes. Automobile prices in market equilibrium. *Econometrica*, 63(4):841 – 890, 1995.

[3] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, September 1999.

[4] A. Boyd and I. Bilegan. Revenue management and e-commerce. *Management Science*, 49(10):1363 – 1386, 2003.

[5] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.

[6] 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(1):242 – 259, 2007.

[7] G. Gowrisankaran and M. Rysman. Dynamics of consumer demand for new durable goods. NBER working paper, 2009.

[8] A. I. Juda, S. Muthukrishnan, and A. Rastogi. Pricing guidance in ad sale negotiations: the printads example. In *ADKDD '09: Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, pages 61–68, New York, NY, USA, 2009. ACM.

[9] S. Muthukrishnan. Ad exchanges : Research issues (to appear). In *Proc of the 4th intl Wkshp on Internet and Network Economics (WINE)*, 2009.

[10] A. Nevo. Measuring market power in the ready-to-eat cereal industry. *Econometrica*, 69(2):307 – 342, 2001.

[11] K. Talluri and G. van Ryzin. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, Norwell, Massachusettes, 2004.

[12] H. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163 – 1178, 2007.

[13] W. E. Walsh, C. Boutilier, T. Sandholm, R. Shields, G. Nemhauser, and D. C. Parkes. Automated Channel Abstraction for Advertising Auctions. In *Proceedings of the Ad Auctions Workshop*, 2009.