Optimize-and-Dispatch Architecture for Expressive Ad Auctions*

David C. Parkes
CombineNet, Inc.
Fifteen 27th St.
Pittsburgh, PA
dparkes@combinenet.com

Tuomas Sandholm
CombineNet, Inc.
Fifteen 27th St.
Pittsburgh, PA
tsandholm@combinenet.com

ABSTRACT

Ad auctions are generating massive amounts of revenue for online search engines such as Google. Yet, the level of expressiveness provided to participants in ad auctions could be significantly enhanced. An advantage of this could be improved competition and thus improved revenue to a seller of the right to advertise to a stream of search queries. In this paper, we outline the kinds of expressiveness that one might expect to be useful for ad auctions and introduce a high-level "optimize-and-dispatch" architecture for expressive ad auctions. The architecture is designed to enable expressiveness while retaining real-time response to search queries.

1. INTRODUCTION

Ad auctions are big business because they provide information about *intentional state* to advertisers. Rather than trying to guess the interests (especially the current focus of interest) of a potential customer, the interaction with a search engine is one in which a user *pushes* information to describe the kind of thing that she is currently interested in learning about. This provides contextual information. For instance, a user might be researching on vacation information related to Costa Rica, or thinking about buying a new laptop. For advertisers this represents the ultimate opportunity in personalized marketing.

Today, the business model of Google is driven by ad auctions, which generated revenue as high as $6,000,000 a day in the third quarter of 2004. Automated bids are submitted on behalf of advisers in response to search queries by a user, with winners gaining the right to display a click-through advert in a panel adjacent to search engine results.[1] Clearly this is a big success. Yet, we wonder: how more more revenue could be generated by finding ways to allow for new bid expressiveness in ad auctions? More expressive bids can lead to better decisions about how to allocate queries to advertisers, and generate more revenue to search engines by promoting more competition.

A lot is known about the design of auctions with expressive bidding languages. Combinatorial auctions (where bids can be expressed on packages of items) are a well-known example. Auctions with richer forms of expressiveness—such as side constraints [16], discount schedules [13, 15, 17], and multi-attribute considerations [16]—have also recently been developed, and tens of billions of dollars of industrial procurement have been conducted with such methods [6, 11, 7].

However, the ad auction problem provides additional challenges:

- First, ads must be served in real time, as quickly as the response to search queries. Moreover, a search engine such as Google is serving millions of responses to queries each day. It is clearly impractical to solve an NP-hard winner-determination problem in the fraction of a second that is available to respond to a query.
- Second, this is an *online* problem in the sense that there is uncertainty both about supply (since query-streams are unpredictable) and about demand (since new bids may be placed, or current bids revoked).
- Third, this is a problem in which *budget constraints* can be expected to be important. The process of bidding in response to queries is necessarily automated to provide adequate speed, and advertisers can be expected to want to place constraints on the total amount they will be billed by the search engine.

We understand that the current state-of-the-art, as implemented by Google, is for an advertiser to submit a bid that defines a maximal willingness to pay for different queries (bid price), together with a budget limit for each day. A winning bid only makes a payment in the event of a click-through on the advert, and because of this the winning bids are those with the highest *expected* payment, as determined through a statistical model of the likelihood of click-through. An alternative model is for payments to be made per-exposure, which can be handled through a simple extension in which a bidder states that this is a "banner"-style ad and to be treated as though each exposure is equivalent to a

*Patent pending.

[1]Yahoo's *Overture* division was the first to implement this business model. Google later adopted a similar model—adding their own modifications —and Google is now licensing the Yahoo/Overture patent. Other companies, such as Microsoft are also pursuing similar business models.

EC'05, June 5–8, 2005, Vancouver, British Columbia, Canada.

click-through. In general, the right to advertise in response to a query can be sold to multiple advertisers, with adverts displayed in a rank order.

The level of expressiveness is quite sophisticated in the ability to define bid prices for various textual queries (based on "good words" and "stop words"). However, the expressiveness is limited in its ability to allow an advertiser to state values for long-term properties of the allocation. For instance, an advertiser might want exclusive rights to some category of queries. Alternatively, an advertiser might care about achieving some minimal level of exposure, by winning some minimal number of auctions.

In this paper, we characterize the kinds of expressiveness that could be used to enhance the efficiency and revenue properties of ad auctions. In addition, we outline an "optimize-and-dispatch" model for how these expressive auctions might be implemented in practice. The main idea is to take high-level optimization decisions in an offline winner-determination engine, that is then used to parameterize a simple online dispatcher. The decisions within the optimizer will be based on a statistical model that predicts query streams. In general, the scheme allows for payments to be determined both within the optimizer and within the dispatcher. There are plenty of details about our proposed architecture that are not formally defined at this point. Rather, our intention is to convey a sense of a new style of approaching the problem. We do this at a level of detail that we believe at least suggests the plausibility of this approach.

Parenthetically, an apparent side benefit of the optimizer module is that it may provide new protection against common forms of Internet fraud that have plagued ad auctions. For instance, when payment decisions are made in the optimizer and when the role of the dispatcher is restricted to the implementation of high-level decisions in the optimizer, one firm cannot drive up the payments made by another firm by clicking on its advert because these payments are defined in the aggregate in the optimizer, based on a statistical model.

1.1 Related Work

The ad auction problem has received some recent attention. For instance, Mehta et al. [10] study a model in which bids express values for different queries and an overall budget constraint. The authors explain how to achieve an optimal worst-case competitive ratio for revenue in this model by adopting a greedy matching algorithm that makes a tradeoff between bid-price and remaining budget. In one of the few papers that has considered incentive-compatibility issues in the context of budget constraints, Borgs et al. [3] study an offline auction problem (although motivated by ad auctions) and characterize restrictions on truthful auctions for a simple multi-unit allocation problem when bidders have budget constraints (that are private information).

2. PRELIMINARIES

In the basic model (for instance as practiced by Google), there is a sequence of queries $Q^1, \ldots, Q^t \in \mathcal{Q}$ that provide supply to the ad auction. Advertiser i can submit bids b_{ij}, that define a *bid-price* $b_{ij}(Q, E)$ for different queries $Q \in \mathcal{Q}$ and different environments $E \in \mathcal{E}$. The environment, E, can be used to capture information such as time-of-day or additional contextual information about a user of the search engine (such as geographical location, sequence of recent click-throughs, time-of-day, etc.). We can think of the bid-price as a statement about the willingness-to-pay of an advertiser.

In general, it is useful to adopt the notion of equivalence classes of queries, with $c \in \mathcal{C}$ denoting a class of queries, such that $c \subseteq \mathcal{Q}$. Then, bid-prices $b_{ij}(c, E)$ can be defined on an equivalence class.

An advertiser can also provide a *budget-constraint*, B_i, which defines the total amount that she will spend in any day. This budget constraint can be further broken down in a number of ways. For instance, an advertiser can provide: a budget-constraint B_{ij} for each bid; a budget-constraint for each bid in a particular class, $B_{ij}(c)$; and an overall budget constraint in each class, $B_i(c)$. For simplicity we assume that these constraints apply in conjunction, so that all must be met in any allocation.

In Google, advertisers only pay when an advert is both displayed and receives a click-through from the user of the search engine. Google maintains a model to estimate the probability $\Pr_{ij}(Q)$ of click-through on advert j from advertiser i given query Q. In the simple case that a single advert is sold in response to a query, i.e. $k = 1$, the query is then sold to the bidder with the maximal *expected* willingness-to-pay $\Pr_{ij}(Q)b_{ij}(Q)$ for a click-through price of $(\Pr(Q)b(Q))^{(2)}/\Pr_{ij}(Q)$ where $(\Pr(Q)b(Q))^{(2)}$ denotes the second-highest expected willingness-to-pay for this query. Thus, the price that the winning advertiser will pay on click-through is set so that her expected payment is equal to the second-highest expected payment. At any time the set of advertisers that compete for queries are those that are still within their budget constraints. We see that the Google ad auction has a Vickrey auction flavor to it, in that a query is sold to the bidder with the highest expected value at a price such that her expected payment is the second-highest expected value.[2]

For our purposes, it will be important to assume the existence of a richer *statistical model*, able to predict the types of queries that are assumed during a day and the numbers of click-throughs that each advert will receive for different classes of queries. It seems quite reasonable to assume the existence of such a statistical model given that a typical search engine sees large volumes of searches each day. We adopt notation M to denote the *model* provided by a search engine, and define expectation $\mathbb{E}_M\{\cdot\}$ with respect to this model. For instance, we will require an estimate of the expected revenue in one day in some query class c given some set of bids, which requires an estimate of the number of queries in this class and the probability of click-through on winning adverts.

3. NEW KINDS OF EXPRESSIVENESS

In describing new forms of expressiveness, we break our presentation into *local* expressiveness and *global* expressiveness. Informally, local expressiveness defines adjustments to bid-price and constraints on outcomes (e.g. "my bid must appear in the top k positions") that can be interpreted and

[2]However, we note that this auction is *not* truthful for a bidder because this is an "online" bidding problem [8, 2, 5, 4, 12, 9, 1]. An advertiser might like to time her bids to be able to advertise at a time of day when there is less competition, because this will tend to reduce her expected payment. For instance, if budgets tend to expire later in the day then a bidder could do better by delaying her bids until later in the day.

implemented based solely on the information local to an auction. On the other hand, global expressiveness defines adjustments to bid-price, bonus payments, and constraints on outcomes (e.g. "I must receive a minimal quantity of exposures for queries in this category in the next month for my bid to be valid") that can only be interpreted in the context of high-level decisions made across a sequence of auctions.

In our framework, global expressiveness is interpreted and acted-upon in the optimizer module (and with a view of local expressiveness in bids), while local expressiveness is interpreted and acted-upon in the dispatcher module. More subtly, our proposed *optimize-and-dispatch* architecture facilities the ability to parameterize the behavior of the dispatch module based on decisions made in the optimizer module, and thus the actual behavior of the dispatch module depends indirectly on the global view that is adopted by the optimizer module and the bid information that is globally expressive. Of course, in order to handle global expressiveness the optimizer module needs access to the statistical model. Moreover, nothing is guaranteed and the dispatcher may not succeed in meeting a constraint (for instance on minimal number of exposures) and thus lose a bonus that was anticipated by the optimizer.

3.1 Global Expressiveness

In this section, we outline some forms of global expressiveness that can be captured in the optimizer module.

Side Constraints

In an expressive ad auction, an advertiser could place constraints on the volume of click-throughs, the level of exposure, and the degree of exclusivity provided over some period of time. These constraints can be interpreted as "these are properties of the allocation over some period of time that must be satisfied for my bid to be valid."

Volume-based Constraints. A bid can be associated with a constraint on the volume of click-throughs. These volume-based constraints may be broken down into different constraints for different search terms, and for different time periods. Consider the following examples:

- a bid might include a MINIMAL side constraint to state the bid is only valid if the bid achieves at least an average of 1000 click-throughs per-day for the period of a campaign.
- a bid might include a MAXIMAL side constraint to state the bid is only valid if the bid achieves no more than an average of 1000 click-throughs per-day for the period of a campaign.[3]

These could also be stated in terms of absolute constraints on click-through over some period of time. Of course, the optimizer can only make recommendations and set targets to the dispatcher, and can only have a statistical belief that targets will be achievable. The dispatcher is designed to try to meet targets set by the optimizer, but some things may not be possible given a realized sequence of queries in any particular day.

An advertiser might also care about "smoothness" constraints, so that volume constraints cannot be satisfied by a rush of adverts over some concentrated period of time. For this, we could also allow an advertiser to express volume constraints on smaller intervals of time. For instance, a bid can include a side constraint to state the bid is only valid if the bid achieves at least 1000 click-throughs in every hour. This can also be relaxed, for example to include a condition to state a relaxed version such as a bid can only violate its volume target in some fraction of time intervals.

Volume-based side constraints might also be stated for *exposure*, that is, the number of times an advert is displayed, in addition to (or instead of) click-through rate. In addition, these constraints can be stated in terms of *eligibility*, which is the notion that the bid was at least eligible to compete for queries for some volume of auctions, or for some fraction of the total number of queries for which the bid expressed some non-zero willingness to pay. In general, constraints could also be broken down into categories of queries in order to provide finer control over the outcomes that are acceptable to an advertiser.

Competition-Based Constraints. A second class of side constraints place restrictions on the level of exclusivity that a bidder requires when her bids are accepted. These constraints can take the following form:

- This bid is valid when it is the only winner, so that the advert does not appear next to any other adverts.
- This bid is only valid if it appears in one of the top N positions in displaying an advert next to search results. (e.g., first position, second position, etc.)
- This bid is only valid if it is the exclusive winning bid in a particular category of search terms for an entire month.
- If this bid wins, then advertisers from set $\mathcal{A}$ must not appear within some rank distance $\pm x$ of the position of my own advert.

Payment Bonuses

Exclusivity-Based. A bid can define a bonus that is available when particular conditions are true about the outcome of a sequence of auctions for queries in which an advertiser has expressed some interest.

- For instance, a bid can state a total additional bonus of $1000 every time Joe's bid is not shown at all for some period of time for a particular advert class.
- For instance, a bid can state a total additional bonus of $5000 when it has the exclusive right to advertise to all queries in a particular class for some period of time.

Volume-Based. A bid can also define a bonus that is available when particular volume targets are met.

- For instance, a bid can state "I will make an additional payment of $100 if at least 100 click-throughs per hour are achieved while my bid is valid."

[3]Such a constraint is similar, but different, to a budget-constraint since the payment made by an advertiser need not be its bid price, and also since the payment made by an advertiser can also include one-time payments. Thus, the two types of constraints are incomparable in their expressiveness in the general ad-auction model that we introduce in this paper.

In cases such as this, the high-level decision about whether or not to seek the bonus is made within the optimizer, although whether or not the bonus is then collected depends on whether or not the dispatcher meets the targets.[4]

These one-time payment bonuses may also be defined in terms of *exposure* and *eligibility* volumes, and might be further refined to include requirements about the smoothness of an allocation across time.

(Global) Click-through Price Adjustments

Global adjustments can also be defined to the click-through price (or bid-price), for instance based on the total volume with which a bid wins an auction or based on the total volume of click-through. Unlike the payment bonuses, this adjustment is made to the click-through price and can be maximized within the dispatcher by finding the allocation that is maximal with respect to the conditions.

For instance, a price schedule can be defined to depend on the volume of click-through (perhaps broken down by search term category and by time of day). This payment schedule can be expressed as a *piecewise linear* function that depends on total volume. For instance, the payment schedule can define an increase in price as the volume increases, such as 0–100 is \$0.00 per click, 100–200 is \$0.05 per click, 200–400 is \$0.07 per click, etc. For smoothness constraints, a bid can be eligible for a 5% increase in bid-price if it is eligible to compete in both the morning and the afternoon periods. Alternatively, a bid might be eligible for a \$0.20 increase in base-price if it is eligible in at least 60% of the auctions in a particular category of search terms. Note that these are described as modifications to the basic bid-price (which can be defined as described below), and made within the optimizer on the basis of a prediction about the allocation statistics.

3.2 Local Expressiveness

In this section, we outline some forms of local expressiveness that can be captured in the dispatch module (and may also factor into the decisions made in the optimizer module).

A Richer Language for Base Prices

In the basic model, an advertiser states a bid-price $b_{ij}(Q)$ for a query Q. Clearly, there is an unbounded number of queries that may be executed on a search engine and so it has been recognized that it is essential to provide a concise language to allow advertisers to express such a bid-price.

Expressing Bids for Queries. First, we sketch some methods to provide for a concise yet expressive method to state a bid-price for different queries $Q \in \mathcal{Q}$. The methods are presented from simple forms of expressiveness to those which are more complex:

1. "Core + Good words, Not Bad words." First, a bid b_{ij} is associated with some set $Core_{ij}$ words, such that when any one word in *Core* is in Q then the base price is initialized to b^0_{ij}. Second, for each additional word in set $Good_{ij}$ up to some limit L_{ij} we add an additional amount b^{add}_{ij} to the base price. Finally, the price is set to zero in the case that any word in set Bad_{ij} is in query Q.

2. "Phrases + Good words, Not bad words." As a slight variation, we can change the first step to require that one of a set of sequences of words, $phrases_{ij}$ is in the query, with the notion of good words and bad words left unchanged.

3. "Class-based, Not bad words." As a further variation, we can suppose that the search defines some semantic classes $c \in \mathcal{C}$ for words and that the core and good word sets are defined by specifying some set of classes that are considered to define these sets of words.

4. "URL based." Perhaps used in combination with one of the other methods, we can also allow the base price to be defined in terms of the URLs that are returned in response to some query. For instance, there can be a set of *core* URLs, *good* URLs and *bad* URLs. The idea behind this approach is to leverage the statistical information implicit in the WWW to guide advertisers in thinking about their value for a query from a user.

5. "Shadow queries." Perhaps used in combination with one of the other methods, we can also imagine that the advertiser's bid can also define a *shadow query*, $s(Q)$, that might strip certain words or add words to the user's query Q, with the idea that executing $s(Q)$ in the background within the search engine might glean further information about the query via the URLs returned. For instance, an advertiser might augment a query with information about an advertiser's business and then gauging the degree of fit between a query and that business from the number of hits returned by the engine.

Expressing Preferences on the User Environment. In addition, an advertiser might wish to express adjustments or restrictions to the bid price, that depend on contextual information about a user over and above that which is implicit in the query. In the preliminaries we captured such additional information within the *environment*, $E \in \mathcal{E}$, noting that a search engine can have additional information about a user's recent search history, geographical location, as well as additional profiling information.

Given this, an advertiser i might want to restrict her bid $b_{ij}(Q)$ to users whose environment $E \in goodenv_{ij}$, where $goodenv_{ij}$ is some class of environments, for instance defined in terms of user demographics and location. Ideally there will be a small set of user demographic and online-usage classifications, from which an advertiser can simply pick out the interesting user types.[5]

Local Click-through Price Adjustments

In all cases, we can also allow an advertiser to express an adjustment to her bid price that depends on local conditions, for instance properties about the context of a user or

[4]This is one way in which discrepancies between the model and reality can be handled. Many variations are possible. For instance, an advertiser could receive a rebate in the case of partially-met targets.

[5]For instance, the time-of-day can be handled by dividing the day into periods such as morning, afternoon, and night and allow separate bid-prices for each period, or restrictions to particular periods. Geographic location can be handled in a similar way, for instance with users in the U.S. divided into aggregated areas based on ZIP code. Given this, then bids can be restricted to particular locations or adjusted based on location.

properties about the rank of the advert, or which adverts are also displayed at the same time:

- For instance, a bid can state an additional per-click-through payment of \$0.20 each time another advert from some particular firm would have won.
- For instance, a bid can state an additional per-click-through payment of \$0.30 each time an advert from another firm is demoted by some number of slots in the rank.
- For instance, a bid can provide an adjustment to the bid-price based on the rank that is received in the auction. For instance, we could allow the advertiser's bid price to decrease by some multiplicative factor that depends on rank, with

$$b_{ij}(Q, m) = \psi_{ij}(m) b_{ij}(Q) \quad (1)$$

  where $m \in \{1, \ldots, k\}$ is the rank of an advert and $\psi_{ij}(m) \geq 0$ is some tradeoff function that re-weights the bid-price $b_{ij}(Q)$ for different ranks. In general, we might expect $\psi(m)$ to be a decreasing function of rank. An advertiser might also state an upper-bound, $maxrank_{ij} \in \{1, \ldots, k\}$ for a bid j, which is the maximal rank that she is willing to accept.
- Adjustments to base price can also be defined for environment conditions, such as time of day, user location, and current user context.

4. THE "OPTIMIZE-AND-DISPATCH" ARCHITECTURE

In this section, we describe the overall *optimize-and-dispatch architecture*. The idea is to perform global (combinatorial) optimization offline in a solver that clears periodically and provides information to parameterize the fast time-of-query decisions that are made within the dispatch module. The optimizer module is designed to handle global expressiveness while the dispatch module is designed to implement high-level decisions made by the optimizer, in addition to handling local expressiveness. Both models make heavy use of a statistical model to provide a distribution over queries and over environments, and to provide the probability of click-through on an advert given a query.

4.1 The Optimizer–Dispatcher Interface

First we define the parameters that can be specified by the optimizer in tuning the behavior of the dispatcher. It is convenient to assume that all targets are defined on a daily basis, although the optimizer itself would need a longer time-horizon for decision making for instance in the case that adverts require exclusivity for some extended period of time such as a month.

It is helpful to sub-divide the parameters into two groups. The first group of parameters defines *overall targets* that the dispatcher should meet during the day and is satisfied via a "throttling" mechanism that controls access to auctions in the dispatcher:

- Budget targets. $\tilde{B}_i(c), \tilde{B}_{ij}(c), \tilde{B}_i$ and $\tilde{B}_{ij}$ define the target budget for advertiser i on queries in class c, for bid j from advertiser i on queries in class c, for advertiser i overall, and for advertiser i on bid j overall.
- Volume targets. $clickfrac_{ij}(c) \in [0, 1]$ defines the fraction of clicks that bid j from advertiser i should win from all clicks in class c. $exposurefrac_{ij}(c) \in [0, 1]$ defines the fraction of adverts that should be awarded to bid j from advertiser i in class c. $eligibilityfrac_{ij}(c) \in [0, 1]$ defines the fraction of adverts for which bid j should be eligible to compete. All of these volume-based considerations can also be defined in absolute, rather than relative, terms.

The second group of parameters are used to modify the precise *rules used to clear any particular auction* in the dispatcher, for instance placing constraints on the number of winners for a particular class:

- Weight $w_{ij}(c, E) \geq 0$. This is the priority given to a bid within the dispatcher, providing an adjustment to the bid-price in determining which bid wins an auction, by class and by environment.
- Reserve prices. $reserve(c, E)$ defines a reservation price for queries in class c, such that no adverts with an expected willingness-to-pay less than this will be displayed and the payments of winning adverts will be at least this in expectation.
- Max number of winners. $max(c, E)$ defines the maximal number of adverts than be displayed in response to any query in class c and for environment E.
- Maximal rank. $maxrank_{ij}(c, E)$ defines the maximal rank that an advert j from i can receive in responding to a query in class c and for environment E.

In fact, we believe that significant power can come from simply restricting the optimizer to specifying weights $w_{ij}(c, E)$ and reserve prices, $reserve(c, E)$. For instance, notice that as a special case this allows the optimizer to provide an advert with an exclusive right to win, by setting its weight to 1 and the weights of other adverts to 0 for some class. As another special case, this allows the adverts that are eligible to compete to be systematically controlled during the day, since a simple environment E can include the local time-of-day of a search engine user.

4.2 The Optimizer Module

The *optimizer module* is executed periodically, and does not need to provide instantaneous responses. It is used to parameterize the dispatcher, by providing eligibility weights $w_{ij}(c, E)$ as well as budget and volume targets, reserve prices, and constraints on the maximal number of adverts that can be displayed for some classes of queries as well as constraints on maximal rank.

Overall, we view this as a hierarchical optimization problem. Let $x \in X$ denote the output of the optimizer, defining all of the information that is used to parameterize the dispatcher (including eligibility weights and targets). Given a set of bids, $bids$, the most general problem can be considered in the following form:

$$\max_{x \in X} \sum_{c \in \mathcal{C}} [\mathbb{E}_M\{rev(c, x_c, bids, \mathbf{q}_c)\} + \mathbb{E}_M\{bonus(c, x_c, bids, \mathbf{q}_c)]$$

$$\text{s.t } x_c \in Feas_M(c, bids), \quad \forall c \in \mathcal{C}$$

$$x \in Feas_M(bids)$$

where $x = (x_1, \ldots, x_C) \in X$ is factored into decisions for each category $c \in \mathcal{C}$ of queries, but there are linking constraints $Feas_M(bids)$ that ensure, for instance, that the overall budget-constraint for any one advertiser is respected in expectation. Notation $Feas_M(c, bids)$ indicates a set of constraints implied by the bids and model M on the allocation x_c on query class c.

The terms in the objective can then be interpreted as follows:

- Revenue $rev(c, x_c, bids, \mathbf{q}_c)$ defines the revenue collected *in the dispatcher* from queries in category c given decision x_c, bids *bids* and given some realized sequence of queries $\mathbf{q}_c$. The optimizer's goal is to maximize the expected revenue, with queries $\mathbf{q}_c$ as predicted in model M. Naturally, the expected revenue depends on the rules of the auction in the dispatcher module (e.g. second-price vs. first-price, etc.).
- Bonus $bonus(c, x_c, bids, \mathbf{q}_c)$ defines the anticipated bonus payment collected *in the optimizer* from queries in class c given decision x_c, bids *bids* and given some realized sequence of queries $\mathbf{q}_c$. Again, this is defined in expectation with respect to model M. The bonus can be broken down into the components defined within the global expressiveness in each bid, for instance to include a bonus for meeting volume targets and exclusivity targets.

Certainly part of the feasibility constraints are related to budget constraints. For instance, given model M and decision x_c^*, and breaking down revenue and bonus to a particular bidder and to a set of bids $j \in \mathcal{B}_i$ submitted by that bidder, we can include the following constraint in $Feas_M(c, bids)$:

$$\gamma_1 \cdot [\sum_{j \in \mathcal{B}_i} \mathbb{E}_M\{rev_{ij}(c, x_c^*, bids, \mathbf{q}_c)\} + \mathbb{E}_M\{bonus_{ij}(c, x_c^*, bids, \mathbf{q}_c)\}] \leq B_i(c), \quad \forall i,$$

where $\gamma_1 \geq 1$ is some parameter to tune how risk-averse the optimizer is in its interpretation of the model. Similar constraints can be expressed for the other possible forms of budget constraints. For instance, the global linking constraints in $Feas_M(bids)$ can include overall budget constraints that are expressed at the bidder level:

$$\gamma_2 \cdot [\sum_{c \in \mathcal{C}} \sum_{j \in \mathcal{B}_i} \mathbb{E}_M\{rev_{ij}(c, x_c^*, bids, \mathbf{q}_c)\} + \mathbb{E}_M\{bonus_{ij}(c, x_c^*, bids, \mathbf{q}_c)\}] \leq B_i, \quad \forall i,$$

where $\gamma_2 \geq 1$ is another parameter to tune how risk-averse the optimizer is in its interpretation of the model, and B_i is used here to denote the overall budget constraint of bidder i.

4.3 The Dispatcher Module

The *dispatcher module* is executed in real time and serves adverts in response to queries. The main control mechanism adopted within the dispatcher is to *throttle* the rate at which each bid can compete for queries, in order to implement the targets specified by the optimizer. Given a query, Q, the basic decision facing the dispatch module, before running a simple auction, is to decide which bids to allow to compete. It is this simplicity of the dispatcher behavior that allows for real-time response and thus enables our optimize-and-dispatch architecture for ad auctions. Once the bids that are eligible to compete for a query are determined, the auction can be cleared, respecting and utilizing information on weighted-eligibility, max-rank position, max-number of winners, bid-price, reservation-price, and budget constraints. For simplicity, we assume that the basic auction adopted by the dispatcher module is a simple variation on the Vickrey-style payments adopted by Google.

The working assumption in the dispatcher module is that any exclusivity constraints or bonuses (or similar global expressiveness) that was defined within bids has already been factored into the decision made by the optimizer, and is captured within the targets and other parameterizations that are passed to the dispatcher. Thus, the only role of the dispatch module is to try to achieve the targets.

Throttling

Let $eligib(Q, E) \subseteq \bigcup_i \mathcal{B}_i$ denote the bids that are eligible to compete in some auction Q given environment E. The dispatcher uses a simple *throttling rate*, $\alpha_{ij}(c, E) \in [0, 1]$ for bid j from advertiser i in query class c and given environment E. This defines the probability with which the bid is eligible to compete. Given query Q and environment E, each bid that is interested in the query (i.e. with some non-zero base price) is eligible to compete with probability $\alpha_{ij}(c, E)$ where $Q \in c$. A simpler version could define some throttling parameter $\alpha_{ij}(c)$ that depends only on the semantic class, or even some α_{ij} that is the same for bidder i across all queries.

Our thought is that one can adopt standard control-theoretic techniques to adjust control parameters α_{ij} to keep the budget, click-through, exposure, and eligibility within some bounds for each bid and for each target class. For instance, one can define bounds on acceptable behavior in tracking a target during a day, and then take corrective measures when the behavior falls outside this acceptable range. Whether or not this simple throttling mechanism is sufficiently powerful in practice, and can meet the targets set within the optimizer, is an important area for practical testing.

Sometimes conflicts might arise between different targets, that were unanticipated within the optimizer. We propose to handle these through a simple prioritization scheme. For instance, we propose that the dispatcher be defined to consider the following prioritization for breaking conflicts:

budget-target $\prec$ *click-through targets* $\prec$ *exposure targets* $\prec$ *eligibility targets.*

The dispatcher first strives to keep within the budget target, and then from all decisions that achieve this chooses that which best meets the click-through targets, and so on.

An Individual Dispatcher Auction

Once the set of competiting bids $eligib(Q, E)$ is determined for a query Q and environment E, these bids are considered within a standard (although budget-modified and weighted) generalization of a Vickrey-style auction.

Consider an example with the following bids, with weights, probability of clickthrough, and bid-price as defined:

bid 1: weight 2, prob 0.1, bid-price \$30
bid 2: weight 1, prob 0.2, bid-price \$20
bid 3: weight 1, prob 0.5, bid-price \$4

Suppose that the method is to auction a single slot (either for reasons of local expressiveness constraints or because of the $\max_k$ constraint from the optimizer.)

Now, the bid with the highest expected weighted bid-price is bid 1, because its expected weighted price is \$6, compared with \$4 and \$2 from bidders 2 and 3. Then, the payment from the winning bidder is $(4 \cdot (1/2))/(0.1)$, which is \$20. This is the second-highest expected weighted payment rescaled by the weights of bidder 1 and 2, and then normalized for the probability of clickthrough on bid 1. The final expected payment is guaranteed to be less than the maximum willingness-to-pay, and shares the same truthfulness properties as the Vickrey-style payments adopted in Google (i.e. without considerations about sequential bidding strategies.)

More generally, we can define the winner-determination problem in the auction as follows:

$$\max_{x_{ik}} \sum_{i=1}^{N}\sum_{k=1}^{M} w_{ij}(c,E)\cdot price_i(k)\cdot x_{i,k} + \sum_{k=1}^{M} x_{0,k} reserve(c,E)$$

$$\text{s.t.} \quad \sum_{i=0}^{N} x_{ik} \le 1, \quad \forall k \le M \tag{2}$$

$$\sum_{i=1}^{N} x_{i,k+1} \le \sum_{i=1}^{N} x_{i,k}, \quad \forall k < M \tag{3}$$

$$\sum_{i=1}^{N} x_{ik} \le 1, \quad \forall k \tag{4}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{M} x_{ik} \le max(c,E) \tag{5}$$

$$x_{ik} = 1, \quad \forall i \ge 1, \forall k > maxrank_{ij}(c,E) \tag{6}$$

$$x_{ik} \in \{0,1\},$$

where x_{ik} indicates whether bid i wins slot k (with a smaller k indicating a higher rank), where $w_{ij}(c,E)$ is the weight as defined for bid j from advertiser i that is relevant for the current query and environment (and similarly for $max(c,E)$ and $maxrank_{ij}(c,E)$). Constant $price_i(k)$ is the expected payment from the bidder if it wins, defined as the *minimal* of the bid-price from advertiser i for rank k (perhaps adjusted both due to global expressiveness and local expressiveness) and the remaining budget, and then multiplied by the probability of click-through. Bidder 0 simulates the role of the reserve price $reserve(c,E)$ for this query, and is willing to buy any number of slots for this price.

Constraints 2 ensure that no slot is sold more than once. Constraints 3 ensure that slots are allocated highest-rank first. Constraints 4 ensure that no bid is allocated more than one slot. Constraints 5 respects the condition from the optimizer that might limit the total number of winners. Constraints 6 respects the limit from the optimizer on the rank that an advertiser is willing to accept. Additional constraints, for instance to provide for exclusivity, or separation to competitors, etc. can also be introduced.

Let $V(N)$ define the revenue with all bids, and $V(N \setminus i)$ define the revenue without bid i. The (expected) generalized Vickrey payments are defined for winners as, $p_{\text{gva},i} =$

$$\frac{1}{w_{ij}(c,E)}\left[V(N\setminus i) - \sum_{i'\neq i}\sum_{k} w_{i'j}(c,E) price_{i'}(k) x^*_{i'k}\right] \tag{7}$$

where x^* denotes the allocation computed in the solution to $V(N)$ with all bids; the final click-through payment is then defined by dividing through by the probability of click-through for bid i.

4.4 Closing the Loop

The idea is to rerun the optimizer periodically, always with the newest information about bids, the newest model of the projected query stream, and the newest information about execution so far (which ads were shown at which ranks at which times, and whether or not they were clicked). The incorporation of the execution history allows the optimizer to in effect monitor the execution and re-parameterize the dispatcher as needed so that the actual execution follows the optimized plan closely.

5. AN ANALOGY TO TRADITIONAL MEDIA BUY

The optimize-and-dispatch architecture has an analog in traditional (TV ad) media buying. In that manual negotiation, the buyers negotiate manually with the sellers in a rather expressive language once a year (in an intense multi-day negotiation conference). That negotiation leads to a high-level plan of how ad time is allocated to the buyers, taking into account time-of-day, viewer segment, and other considerations. The execution over the year does not exactly follow the plan because shows get canceled and introduced, there are outages, etc. To mitigate that online aspect, the TV broadcasters dispatch ads as they best see fit so as to try to roughly adhere to the overall plan for the year. (At the end of the year, significant discrepancies are addressed with rebates.)

6. CONCLUSIONS AND FUTURE RESEARCH

Ad auctions are generating massive amounts of revenue for companies such as Google with online search engines. Yet, the level of expressiveness provided to participants in ad auctions could be significantly enhanced. An advantage of this could be improved competition and thus improved revenue to a seller of the right to advertise to a stream of search queries. This benefit does not necessarily come from making the buyers worse off: while the buyers end up potentially paying more, they have better control and targeting of their marketing campaigns. In this paper, we outlined the kinds of expressiveness that one might expect to be useful for ad auctions and introduced a high-level "optimize-and-dispatch" architecture for expressive ad auctions. The architecture is designed to enable expressiveness while retaining real-time response to search queries.

Future research includes implementing a system based on this architecture—and using (generalizations of) state-of-the-art commercial winner determination algorithms [14] within the optimizer module. It will then be exciting to see 1) which expressiveness forms the buyers will find useful, 2) how much the increased expressiveness improves the auctions, and 3) how often the optimizer needs to be re-run in practice so as to keep the execution closely enough aligned with the optimized plan (clearly there is a tradeoff between supporting rich expressiveness leading to slower optimization, and following the plan more closely by running the optimizer more often).

7. REFERENCES

[1] B. Awerbuch, Y. Azar, and A. Meyerson. Reducing truth-telling online mechanisms to online optimization. In *Proc. ACM Symposium on Theory of Computing (STOC'03)*, 2003.

[2] A. Blum, T. Sandholm, and M. Zinkevich. Online algorithms for market clearing. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 971–980, San Francisco, 2002.

[3] C.Borgs, J.Chayes, N.Immorlica, M.Mahdian, and A.Saberi. Multi-unit auctions with budget-constrained bidders. In *Proc. ACM Conf. on Electronic Commerce*, 2005.

[4] M. T. Hajiaghayi, R. Kleinberg, M. Mahdian, and D. C. Parkes. Online auctions with re-usable goods. In *Proc. ACM Conf. on Electronic Commerce*, 2005. To appear.

[5] M. T. Hajiaghayi, R. Kleinberg, and D. C. Parkes. Adaptive limited-supply online auctions. In *Proc. ACM Conf. on Electronic Commerce*, pages 71–80, 2004.

[6] G. Hohner, J. Rich, E. Ng, G. Reid, A. J. Davenport, J. R. Kalagnanam, H. S. Lee, and C. An. Combinatorial and quantity-discount procurement auctions benefit Mars, incorporated and its suppliers. *Interfaces*, 33(1):23–35, 2003.

[7] R. Hughes, J. Jacobs, D. Begg, T. Sandholm, D. Levine, and M. Concordia. Changing the game in strategic sourcing at Procter & Gamble: Expressive competition enabled by optimization. *Interfaces*, 2005. Edelman award competition finalist writeup.

[8] R. Lavi and N. Nisan. Competitive analysis of incentive compatible on-line auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, 2000.

[9] R. Lavi and N.Nisan. Online ascending auctions for gradually expiring goods. In *In Proc. SODA'05*, 2005.

[10] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized on-line matching. Technical report, Georgia Tech, 2005.

[11] T. Metty, R. Harlan, Q. Samelson, T. Moore, T. Morris, R. Sorensen, A. Schneur, O. Raskina, R. Schneur, J. Kanner, K. Potts, and J. Robbins. Reinventing the supplier negotiation process at Motorola. *Interfaces*, 35(1):7–23, 2005.

[12] R. Porter. Mechanism design for online real-time scheduling. In *Proc. ACM Conf. on Electronic Commerce (EC'04)*, 2004.

[13] T. Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence*, 18(4):656–676, 2002. Special issue on Agent Technology for Electronic Commerce. Early versions appeared in the Conference on Autonomous Agents (AGENTS-00), pp. 73–96, 2000; AAAI-99 Workshop on AI in Electronic Commerce, Orlando, FL, pp. 46–55, July 1999; and as a Washington University, St. Louis, Dept. of Computer Science technical report WU-CS-99-02, Jan. 1999.

[14] T. Sandholm. Winner determination algorithms. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*. MIT Press, 2005.

[15] T. Sandholm and S. Suri. Market clearability. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1145–1151, Seattle, WA, 2001.

[16] T. Sandholm and S. Suri. Side constraints and non-price attributes in markets. In *IJCAI-2001 Workshop on Distributed Constraint Reasoning*, pages 55–61, Seattle, WA, 2001. To appear in *Games and Economic Behavior*.

[17] T. Sandholm and S. Suri. Optimal clearing of supply/demand curves. In *13th Annual International Symposium on Algorithms and Computation (ISAAC)*, Vancouver, Canada, Nov. 2002. Also appeared in the proceedings of the AAAI-02 workshop on Agent-Based Technologies for B2B Electronic Commerce (AAAI Technical Report WS-02-01), pp. 15–22, Edmonton, Canada.