Automated Channel Abstraction for Advertising Auctions*

William E. Walsh
CombineNet, Inc.
Fifteen 27th St.
Pittsburgh, PA, USA
wwalsh@combinenet.com

Craig Boutilier
Dept. of Computer Science
University of Toronto
Toronto, ON, CANADA
cebly@cs.toronto.edu

Tuomas Sandholm
Computer Science Dept.
Carnegie Mellon University
Pittsburgh, PA, USA
sandholm@cs.cmu.edu

Rob Shields
CombineNet, Inc.
Fifteen 27th St.
Pittsburgh, PA, USA
shields@combinenet.com

George Nemhauser
ISYE
Georgia Inst. of Technology
Atlanta, GA, USA
gnemhaus@isye.gatech.edu

David C. Parkes
SEAS
Harvard University
Cambridge, MA, USA
parkes@eecs.harvard.edu

Abstract

The use of simple auction mechanisms like the GSP in online advertising can lead to significant loss of efficiency and revenue when advertisers have rich preferences—even simple forms of expressiveness like budget constraints can lead to suboptimal outcomes. While the optimal allocation of inventory can provide greater efficiency and revenue, natural formulations of the underlying optimization problems grow exponentially in the number of features of interest, presenting a key practical challenge. To address this problem, we propose a means for automatically partitioning inventory into *abstract channels* so that the least relevant features are ignored. Our approach, based on LP/MIP column and constraint generation, dramatically reduces the size of the problem, thus rendering optimization computationally feasible at practical scales. Our algorithms allow for principled tradeoffs between tractability and solution quality. Numerical experiments demonstrate the computational practicality of our approach as well as the quality of the resulting abstractions.

Introduction

Online advertising has radically changed the nature of advertising and the technology supporting the deployment of ad campaigns. While ad targeting and campaign design is inherently complex, the variety of online advertising services has only increased this complexity. In particular, the ability to target ads to *specific individuals* based on detailed, personalized online information—information that is simply not available in broadcast media—presents compelling opportunities and tremendous technical challenges for ad delivery. Sophisticated matching and bidding algorithms, such as auctions using *generalized second price (GSP)* (Edelman, Ostrovsky, & Schwarz 2007; Varian 2007), have been developed for sponsored search advertising. By contrast, the selling of graphical *display ads* on web pages is still largely managed via manual negotiation. Though much low-value *remnant* inventory is sold in online exchanges, *premium* display advertising space (e.g., slots near the top, or "above the fold," of high traffic, high profile websites) is sold almost exclusively by non-automated means. One reason for this is a perception that auction/market mechanisms cannot be made to work for the types of *campaign-level expressiveness* (e.g., impression targets, smoothness of delivery, temporal sequencing, complements, representativeness) required for display ads (Parkes & Sandholm 2005; Boutilier *et al*. 2008).

While sophisticated bidding strategies (Borgs *et al*. 2006; Feldmann *et al*. 2007; Rusmevichientong & Williamson 2006) can increase the value a bidder extracts from an inexpressive auction (e.g., GSP) for some limited preference types (e.g., long-term budgets), it is very difficult to bid effectively with more demanding types of preferences (e.g., requiring minimum quantities). Furthermore, with inexpressive auctions, arbitrarily large inefficiencies can arise in general (Benisch, Sadeh, & Sandholm 2009). Richer languages that allow advertisers to express their true campaign preferences directly, rather than forcing them into standard per-event bidding models, are critical to the automated matching and selling of display ads. It is just these forms of *campaign-level expressiveness* that are developed in (Parkes & Sandholm 2005; Boutilier *et al*. 2008), where a variety of expressiveness forms are outlined. But a significant bottleneck remains: the use of expressive bidding requires *optimization* to match ad supply with advertisers' demand.

In this paper we tackle a key impediment to the use of optimization in ad auctions: *channel explosion*. Online advertisers can segment the target audience and ad impressions using an enormous number of features. But the number of *ad channels*, or feature instantiations, to which ads can be assigned grows exponentially in the number of features. Standard models that use linear programming (LP) (Abrams, Mendelevitch, & Tomlin 2007) or mixed-integer programming (MIP) (Boutilier *et al*. 2008; Parkes & Sandholm 2005) to assign ads to such channels simply cannot scale directly to problems involving more than a few thousand channels. We address this through the use of *channel ab-*

*This work was sponsored by, and conducted at, CombineNet, Inc. Patent pending.

straction. Intuitively, an abstract channel is any aggregation of concrete channels (i.e., feature instantiations) into a single channel. During allocation optimization, ads are assigned to abstract channels rather than concrete channels. As we show, a well-chosen abstraction, guided by its impact on *allocation value*—as opposed to clustering based solely on statistical properties of the features in question—can make optimization practical with little sacrifice of revenue or efficiency. We propose techniques for automatically generating and using a set of abstract channels: a novel form of *column generation* to generate an abstraction; and a new *constraint generation* algorithm for improving the allocation of ads to abstract channels.

In the next section we present the basic ad allocation model and define abstract channels. We show that a small number of channels is sufficient to implement an optimal allocation. We then develop a novel and computationally effective column generation technique to generate useful abstractions—empirical results show that the algorithm obtains near-optimal allocations with very few channels. We extend the approach with a constraint generation algorithm that makes more effective use of abstract channels, and demonstrate how it significantly improves value when "MIP expressiveness" is involved. We conclude with directions for future research.

Allocation Model and Abstract Channels

We assume a finite attribute or *feature* set $\mathcal{F}$, each $F^i \in \mathcal{F}$ having finite domain $Dom(F^i) = \{f^i_1, \ldots, f^i_{n^i}\}$. Features describe attributes of an ad display such as web site, page location, user demographic, day part, contextual features, etc. The set of *concrete channels (c-channels)* C comprises the instantiations of features $\mathcal{F}$. Intuitively, a c-channel $c \in C$ is a finest-grained chunk of supply to which an ad can be assigned. We often treat c as a model of the propositional language over variables $\mathcal{F}$ (e.g., writing $c \models \varphi$ for propositional formulae φ over $\mathcal{F}$). Let $s(c, t)$ be the supply of c-channel c available at time $t \leq T$.

Advertisers express their campaign objectives using a set of one or more bids, potentially linked by shared variables and constraints. While we allow all forms of expressiveness that can be represented as a MIP, we motivate our techniques using a simple LP-based model. We assume a set $\mathcal{B}$ of item-based, budget-constrained bids. Each bid $i \in \mathcal{B}$ has form $\langle \varphi^i, v^i, g^i, w^i \rangle$, where φ^i is a logical formula over features $\mathcal{F}$, $v^i > 0$ is i's price per impression, $g^i > 0$ is its budget, and w^i is a time window $[s^i, e^i]$ within which impressions must occur ($1 \leq s^i \leq e^i \leq T$). Bid i reflects advertiser i's interest in impressions satisfying the condition φ^i. The allocation problem in this setting can be formulated as a simple LP that maximizes revenue by allocating $x^i_c(t)$ impressions of c-channel $c \in C$ to bid i at time t. To simplify notation, we formulate the optimization as if there were a single time period. (The generalization to multiple periods is obvious). Let v^i_c be i's value for a c-impression: $v^i_c = v^i$ if $c \models \varphi^i$; $v^i_c = 0$ otherwise. Then we have (with $x^i_c \geq 0$):

$$\max_{x^i_c} \quad \sum_i \sum_c v^i_c x^i_c \tag{1}$$
$$\text{s.t.} \quad \sum_i x^i_c \leq s(c) \quad \forall c \in C; \quad \sum_c v^i_c x^i_c \leq g^i \quad \forall i \in \mathcal{B}.$$

This LP can easily be extended to other forms of *LP expressiveness*, such as substitutes, complements, and time-based smoothness. For example, if a campaign has (partially) substitutable demands (e.g., it desires φ_1 or φ_2 with values v_1 and v_2), two separate bids can be posted with a joint budget constraint. If φ_1 and φ_2 are complements, we can constrain the allocated impressions to meet some approximate ratio target (e.g, $imp(\varphi_1) \leq (1+\varepsilon) imp(\varphi_2)$, $imp(\varphi_2) \leq (1+\varepsilon) imp(\varphi_1)$, where $imp(\varphi)$ is the number of impressions of φ). Smoothness constraints can also be encoded linearly (e.g., requiring at least 10% of total impressions to be allocated in each eligible time period). A bidder may want to receive a "representative allocation" (Ghosh *et al.* 2009), whereby the distribution of the attributes of impressions received reflects that in the overall population matching φ^i. Bidders may also want to cap the frequency that an ad is shown to any given user. We can model all of these forms of bid expressiveness within the LP.

Other forms of *MIP expressiveness* requires the use of binary variables, for example, threshold/bonus bids in which an advertiser requires a certain minimum quantity of impressions (Boutilier *et al.* 2008; Parkes & Sandholm 2005). Our model also generalizes readily to per-click and per-action valuation. For a deeper discussion of expressiveness forms, see (Boutilier *et al.* 2008).

Abstract Channels The number of c-channels $|C|$ grows exponentially in the number of features. Thus we must consider the use of *abstract channels (a-channels)*. An abstract channel is any aggregation of c-channels, and can be represented as a logical formula α over $\mathcal{F}$. An *abstraction* is a partitioning of c-channels C into a set A of a-channels, i.e., a set of mutually exclusive and covering formulae $\{\alpha_1, \ldots, \alpha_{|A|}\}$. We treat an a-channel and its logical representation α indistinguishably, writing both $c \in \alpha$ and $c \models \alpha$ as appropriate. *Lossless abstraction* is one means of creating a-channels: we group c-channels corresponding to (logically consistent) formulae of the form $\wedge_{i \in \mathcal{B}} \pm \varphi^i$; i.e., conjunctions over all bid formulae or their negations. While this allows for optimal allocation, it it will not generally lead to a manageable number of channels; instead we consider "approximation" using a-channels that are not necessarily aligned with bid formulae.

Given an abstraction A, our optimization problem is one of assigning bids to *a-channels*. Define the supply of a-channel α to be $s(\alpha) = \sum\{s(c) : c \in C, c \models \alpha\}$. We formulate the optimization assuming a *random dispatch policy*: if i is assigned to an abstract channel α, it's ad will be dispatched randomly to the c-channels that constitute α. Under this assumption, the probability that an α-impression is relevant for bid i is $p^i_\alpha = Pr(\varphi^i|\alpha)$, where $\Pr(\varphi^i|\alpha) = s(\varphi^i \wedge \alpha)/s(\alpha)$. Thus, for channel α, the number of specific impressions out of x^i_α that "count towards" the satisfaction of a bid i's conditions is $p^i_\alpha x^i_\alpha$. In particular, for our simple LP, the value of a single α-impression to i is $v^i_\alpha = v^i p^i_\alpha$. This reflects the (expected) value of a *random dispatch policy*: if i is assigned to an abstract channel α, it will be assigned randomly to the c-channels that constitute α.[1] The optimal allocation under the random dispatch assumption is given by the LP:

$$\begin{array}{lll} \max\limits_{x^i_\alpha} & \sum\limits_i \sum\limits_\alpha v^i p^i_\alpha x^i_\alpha & \\ \text{s.t.} & \sum\limits_i x^i_\alpha \le s(\alpha) & \forall \alpha \in A \\ & \sum\limits_\alpha v^i_\alpha x^i_\alpha \le g^i & \forall i \in \mathcal{B}. \end{array}$$

With more general expressiveness, we may not associate value directly with impressions, but with complex properties of the entire allocation, possibly involving multiple formulae φ^i. In such a case, we discount the impressions that count toward satisfaction of the any component formula φ^i by $\Pr(\varphi^i|\alpha)$. The value discount in the per-impression LP is a special case of this.

We wish to obtain an abstraction that allows optimization to tractably achieve a high-value allocation. Fortunately, such an allocation always exists.

Theorem 1 *For any abstraction A with an allocation Υ in which W is the set of bids with positive allocation, there exists an abstraction A' with a corresponding allocation Υ' such that $|A'| \le 2|W| - 1$ and each bid receives the same number of relevant impressions in Υ' as in Υ.*

Proof sketch. Construct a bipartite graph $G = \{V_B, V_C, E\}$ with a *bid vertex* for each $i \in W$ and a *channel vertex* for each $\alpha \in A$. An edge with weight x^i_α exists between a bid vertex i and a channel vertex α iff there is a positive allocation from channel α to bid i in the allocation Υ.

If any cycles exist in G, choose one and consider the subgraph $G' = \{V'_B, V'_C, E'\}$ comprising the edges in this cycle. Let $k = |V'_B| = |V'_C|$, and label the bid and channel vertices from 1 to k, s.t. bid vertex 1 connects to channel vertices 1 and 2, etc. We break this cycle by *shifting* the allocation, holding each bid's relevant impression total constant and not exceeding the supply used in each channel. This corresponds to finding a new set of (non-negative) edge weights $\overline{x}$, with at least one $\overline{x}^i_\alpha = 0$, satisfying:

$$\begin{gathered} \overline{x}^1_1 + \overline{x}^k_1 \le x^1_1 + x^k_1 \\ \cdots \\ \overline{x}^{k-1}_k + \overline{x}^k_k \le x^{k-1}_k + x^k_k \\ \\ p^1_1 \overline{x}^1_1 + p^1_2 \overline{x}^1_2 = p^1_1 x^1_1 + p^1_2 x^1_2 \\ \cdots \\ p^k_k \overline{x}^k_k + p^k_1 \overline{x}^k_1 = p^k_k x^k_k + p^k_1 x^k_1 \end{gathered}$$

This system must have a solution with $\overline{x}^i_\alpha = 0$ for some i, α. We update the graph G by changing all edge weights x in the cycle to these new weights $\overline{x}$ and remove the edge with (new) weight 0. We repeat this process until G is acyclic.

Channel vertices with degree 1 in the new graph (*singleton channels*) are those in which only one bid receives positive allocation. Any two singleton channels α and β allocated to the same bid i can be collapsed into a single channel $\alpha \vee \beta$, while preserving total relevant impressions, as follows:

$$x^i_{\alpha\vee\beta} = \frac{p^i_\alpha x^i_\alpha + p^i_\beta x^i \beta}{p^i_{\alpha\vee\beta}}.$$

We then replace α and β by a single new channel vertex $\alpha \vee \beta$ with a single edge of weight $x^i_{\alpha\vee\beta}$ connected to the bid vertex i. Channels not allocated to a bid can be collapsed into any such singleton channel by the same process.

By maximal collapse of singletons, we have a bipartite graph with at most $|W|$ channel vertices of degree 1 (or 0). Since the graph is now acyclic, there are at most $|W| - 1$ channel vertices with degree 2 or more. Hence there are at most $2|W| - 1$ channel vertices in the new graph. Construct an abstraction A' and allocation Υ' corresponding to the final graph. Each operation preserved the total of relevant impressions awarded to each bid, and $|A'| \le 2|W| - 1$. □

It immediately follows that an optimal allocation requires at most $2|W| - 1$ channels. The proof is constructive given the initial allocation, but does not provide any guidance for how to come up with an optimal initial allocation. In the following sections we describe instead how to generate small, high-quality abstractions based on column generation.

Creating Abstractions: Column Generation

The solution of an abstract LP or MIP (depending on the form of expressiveness allowed in the market) provides us with an optimal assignment of bids to a-channels. This leaves the question of choosing a set of a-channels of computationally-manageable size, yet whose solution provides a near-optimal solution to the original problem. We develop a novel column generation method to do just this. We first describe the method using LPs with only supply constraints, then show how it applies more broadly to arbitrary LP and IP expressiveness.

The basic approach is as follows: we solve an abstract LP using a trivial initial abstraction (e.g., aggregating all channels into a single a-channel $\top$). We refine the abstraction by splitting an a-channel α by conjoining a formula β and its negation, thus replacing α by $\alpha \wedge \beta$ and $\alpha \wedge \overline{\beta}$. A new LP is solved with the new a-channels, and the process repeats until the improvement in LP objective value falls below some threshold or the number of channels reaches a specified limit. To illustrate, consider an LP to allocate a single a-channel α to bids $\mathcal{B} = \{1, 2\}$ (with no bid constraints):

$$\begin{array}{llll} \max & v^1_\alpha x^1_\alpha & +v^2_\alpha x^2_\alpha & \\ \text{s.t.} & x^1_\alpha & +x^2_\alpha & \le s(\alpha). \end{array}$$

and $x^1_\alpha, x^2_\alpha \ge 0$. Refining α requires introducing the bid columns (and supply rows) corresponding to $\alpha \wedge \beta, \alpha \wedge \overline{\beta}$

[1] The dispatch of ads can be handled more intelligently: no ad for i will actually be assigned to a channel not satisfying φ^i; intelligent dispatch (Parkes & Sandholm 2005) can be used to reassign such wasted supply to ads that can exploit it. Thus, v^i_α underestimates true value. We discuss this below, and develop methods to assign ads to abstract channels in a more refined fashion.

for some β. We first discuss how to evaluate the quality of candidate βs, and then how to search for the best split.

Scoring Abstract Channel Splits

The process of splitting α by β requires introducing new columns (variables) to the LP. *Column generation* (Lübbecke & Desrosiers 2005) is widely used to solve LPs with very large numbers of columns by first solving a version of the LP with few columns, then adding new columns at each iteration and resolving. New columns are chosen by solving a *pricing subproblem* which identifies columns that potentially improve the objective. We adopt this approach, but require significant enhancements that exploit the special structure of our problem, and account for the introduction of multiple columns at once (i.e., $x^i_{\alpha\wedge\beta}$ and $x^i_{\alpha\wedge\overline{\beta}}$ for each bid i) while simultaneously removing other columns (i.e., those for x^i_α).

Assume we have the solution of the abstract LP above. We determine the value, or *score*, of a potential split of α into two a-channels $\alpha\wedge\beta, \alpha\wedge\overline{\beta}$ by: (a) scoring the new columns introduced by the split using a form of column generation scoring; and (b) combining the scores of these new columns in a way that exploits the special structure of our problem. Standard column generation methods solve the pricing subproblem to identify individual columns absent from an LP with positive *reduced cost* and typically add one or more such columns with high reduced cost, terminating when no reduced costs are positive. We apply a similar technique. Let π_α be the value of the dual variable corresponding to the supply constraint for a-channel α in the dual of the abstract LP (i.e., the shadow price of the constraint). The reduced cost of variable $x^i_{\alpha\wedge\beta}$ is:

$$rc(x^i_{\alpha\wedge\beta}) = v^i_{\alpha\wedge\beta} - \mathbf{c}\pi,$$

where $\mathbf{c}$ is $x^i_{\alpha\wedge\beta}$'s column (i.e., the vector of coefficients for $x^i_{\alpha\wedge\beta}$ over the rows) and π is the vector of dual variables over the rows. The reduced cost of $x^i_{\alpha\wedge\overline{\beta}}$ is defined similarly. Reduced cost measures the increase in objective value per unit increase in the (nonbasic) variable, making maximum reduced cost a common, easily computable *heuristic* for variable introduction. (It can also be used to prove optimality when max reduced cost is nonpositive.) Although $\mathbf{c}\pi$ measures the marginal impact of constraints w.r.t. the variable, reduced cost is a heuristic since it fails to consider how far the target variable can be moved until constraints are met.

Unfortunately, the abstract LP does not include relevant supply constraints for $\alpha\wedge\beta$ or $\alpha\wedge\overline{\beta}$, meaning shadow prices cannot be directly obtained from the LP. If we add two rows to the abstract LP reflecting split channel supply, we obtain:

$$\begin{array}{llll}
\text{Max} & v^1_\alpha x^1_\alpha & +v^2_\alpha x^2_\alpha & \\
\text{s.t.} & x^1_\alpha & +x^2_\alpha & \leq s(\alpha) \\
 & \Pr(\beta|\alpha)x^1_\alpha & +\Pr(\beta|\alpha)x^2_\alpha & \leq s(\alpha\wedge\beta) \\
 & \Pr(\overline{\beta}|\alpha)x^1_\alpha & +\Pr(\overline{\beta}|\alpha)x^2_\alpha & \leq s(\alpha\wedge\overline{\beta}).
\end{array}$$

Since $s(\alpha\wedge\beta) = \Pr(\beta|\alpha)s(\alpha)$ (similarly for $\overline{\beta}$), the new constraints are multiples of the $s(\alpha)$ constraint, leaving the optimal solution unaffected. This allows us to price the two new constraints: when we consider the dual of this LP, one optimal solution sets the dual variable π_α to its value in the original abstract dual LP, and sets the two new dual variables $\pi_{\alpha\wedge\beta} = \pi_{\alpha\wedge\overline{\beta}} = 0$. As a result, we can compute the reduced costs of the split channel variables using terms available from the solution of the original abstract LP:[2]

$$\begin{aligned}
rc(x^i_{\alpha\wedge\beta}) &= v^i_{\alpha\wedge\beta} - \mathbf{c}\pi = v^i_{\alpha\wedge\beta} - \pi_\alpha \\
rc(x^i_{\alpha\wedge\overline{\beta}}) &= v^i_{\alpha\wedge\overline{\beta}} - \mathbf{c}\pi = v^i_{\alpha\wedge\overline{\beta}} - \pi_\alpha.
\end{aligned}$$

In contrast to typical column generation, we want to model the impact of simultaneously introducing the entire *set* of new columns created by a split, and *removing* the entire set of columns corresponding to the original channel. Nevertheless, reduced cost forms the basis of an effective scoring function. With only supply constraints, we can measure the *exact* change in objective value resulting from a split. If bids have no budget constraints, all supply of the new split channel $\alpha\wedge\beta$ will be allocated to the bid i that has maximum value $v^i_{\alpha\wedge\beta}$, giving objective value improvement of $rc(x^i_{\alpha\wedge\beta})s(\alpha\wedge\beta)$. Here the reduced cost component reflects the precise difference in objective value if an α-impression to a current winning bid is replaced by an $\alpha\wedge\beta$-impression to bid i, while the supply component tells us exactly how much substitution is possible. Applying the same argument to $\alpha\wedge\overline{\beta}$ gives the following *score* for the split of any α into two subchannels $\alpha\wedge\beta$ and $\alpha\wedge\overline{\beta}$:

$$\begin{aligned}
\mathit{score}(\alpha,\beta,\overline{\beta}) = &\max_{i\in\mathcal{B}}\{rc(x^i_{\alpha\wedge\beta})s(\alpha\wedge\beta)\} \\
&+ \max_{i\in\mathcal{B}}\{rc(x^i_{\alpha\wedge\overline{\beta}})s(\alpha\wedge\overline{\beta})\}.
\end{aligned}$$

This scoring function has the desirable property that the score of a split is *exactly* the induced improvement in objective value when only supply constraints are present. Of course, almost all problems have other constraints (budget, etc.), which would be accounted for appropriately in the reduced cost calculation. Still, the reduced cost calculation remains straightforward for LP expressiveness, requiring only one vector product (using dual values computed in the LP solution). Moreover, the score provides an upper bound on possible objective value improvement, and a guarantee of optimality if the maximum score is nonpositive, even when other constraints are present.[3] A key advantage of our scoring function is that no additional computation is required apart from reduced cost calculations (using terms available from the LP solve) and a trivial maximization. This is critical, since the number of potential splits is doubly exponential, as discussed next.

Searching for Suitable Splits

Scoring a split is straightforward, requiring at most $2|\mathcal{B}|$ reduced cost calculations. However, the number of potential splits of an a-channel is doubly exponential in n (i.e., 2^{k^n} formulae over n features with domain size k). In addition, we must evaluate splits of each α in the current abstraction

[2] For more general expressiveness, we would also subtract $c_r\pi_r$, for any non-supply constraint r.

[3] One could use more complex, computationally demanding scoring to better estimate objective improvement, but folklore in column generation suggests this is rarely worthwhile.

A. To manage the complexity of this search, we adopt a simple myopic approach to find the best split of an a-channel α. We build up the formula β_α on which α is split as follows. Let $\overline{f_k^i} = Dom(F^i) \setminus \{f_k^i\}$. We first consider each β_α^1 consisting of $\overline{f_k^i}$ for some i, k; i.e., at the first "level" we consider splits that exclude one attribute-value. We "commit" to the single attribute-value exclusion with the best score $score(\alpha, \beta_\alpha^1, \overline{\beta}_\alpha^1)$. We then consider refining β_α^1 by conjoining with some new $\overline{f_k^i}$ or disjoining with some new f_k^i (conjoining tightens β_α^1, disjoining relaxes it). Each resulting β_α^2 is scored in a similar fashion, and we again commit to the β_α^2 with the highest score. This continues for ℓ iterations, where ℓ is either a fixed threshold or is determined dynamically by requiring a minimum score improvement be met. The best split of α is determined heuristically as $\langle \beta_\alpha, \overline{\beta}_\alpha \rangle$, where $\beta_\alpha = \beta_\alpha^\ell$.

Given abstraction A, the $\alpha \in A$ with the highest-scoring best split is adopted, creating a new abstraction A' with α replaced by $\alpha \wedge \beta_\alpha$ and $\alpha \wedge \overline{\beta}_\alpha$. The LP for the new abstraction is solved and the search for a best split repeated until the score of the best split of A falls below some threshold τ.

Using Abstractions in Ad Auction Optimization

A limitation of our column generation method as specified is its focus on LP expressiveness. However, the abstraction process is used to create the set of a-channels which are then *used* in MIP optimization—the intended output is a set of a-channels, not (necessarily) the allocation itself. With MIP expressiveness, we apply column generation to a linear relaxation of the MIP to generate a-channels. We then solve the original MIP using allocation to the a-channels created. To evaluate column generation, we ran it on a collection of random problems, some with LP expressiveness, others with MIP expressiveness. All experiments were run on a machine with a 3.8GHz Xeon CPU, 2BM cache, and 16GB RAM.

LP Expressiveness The first battery of problems involves bids that use only LP expressiveness, each having per-impression valuations for a set of attribute-values over a given period, and a total budget. Optimization is performed over a horizon of 30 periods. Problem instances are characterized by parameter m: m binary attributes and $10m$ bidders. We run sets of instances with $m \in \{10, 20, \ldots, 100\}$.

Supply distribution. The probability of an impression satisfying f_1^i is drawn from $U[0, 1]$ and we set $\Pr(f_2^i) = 1 - \Pr(f_1^i)$. Total supply of impressions, over all attribute-values, is 1,000,000 per period.

Bids. Each bid j has form $\langle \varphi^j, v^j, g^j, w^j \rangle$ and cares about a set of attributes A^j with size $|A^j| \sim U[0, 10]$. We assume bidders tend to care about similar attributes, so bid attributes are sampled from a Zipf distribution, with $\Pr(F^i \in A_j) = (1/i)/(\sum_{k=1}^m 1/k)$, sampled without replacement. For any $F^i \in A^j$, bid j requires that impressions satisfy $f_{z_i}^i$, with $z_i \in \{1, 2\}$ chosen uniformly. The bid's formula is the conjunction of all required attributes, $\varphi^j = \bigwedge_{F^i \in A^j} f_{z_i}^i$.

Our bid valuation model assumes that higher values are more likely for specific bids (i.e., with more attributes) and if the attributes in the bid formula are in greater demand. Bid j's per-impression value v^j is determined thusly: we first draw a *base value* $\hat{v}^j$ from $U[0.1, 1]$ then adjust it by setting $v^j = \hat{v}^j(1 + 10 \sum_{F^i \in A^j} \Pr(F^i))$ (e.g., if a bid cares about no attributes, i.e., $\varphi^j = \top$, then $v^j = \hat{v}^j$; and if it cares about all m attributes, then $v^j = 11\hat{v}^j$). A bid's time window w^j is determined by sampling t_1 and t_2 from $U[-10, 40]$, setting $w^j = [\min(t_1, t_2), \max(t_1, t_2)]$, then truncating w^j to lie in $[1, 30]$. This captures the fact that some bids have windows that extend beyond the optimization horizon. Bid j's budget is set to a fraction $\tau^j \sim U[0.1, 1]$ of its value for the total supply σ^j in window w^j of the formula φ^j it desires: $g^j = \tau^j \sigma^j v^j$.

In addition to these bids, we include a "market" bid with value 0.1, unlimited budget, and no attribute preferences (i.e., $\varphi = \top$), reflecting value that could be obtained from other sources (e.g., future bids or a spot market).

Optimization parameters. During an iteration of column generation, we continue searching for a suitable split as long as we can find a channel refinement whose score offers a minimum relative improvement *MI* over the previous abstraction's LP value. If such an improvement is found, we solve the new abstract LP and iterate, otherwise we terminate column generation.[4]

Estimating an upper bound on the optimal value. To measure how good an allocation is, we need to estimate the true optimum value achievable if we generated all relevant columns. We compute an upper bound on the optimum as follows. When column generation is complete, we run another optimization using *undiscounted* values. That is, we remove all $\Pr(\varphi^i|\alpha)$ terms. This is clearly an upper bound on the optimum because it assumes that bids could actually make use of the entire amount of a channel it is allocated (rather than just the fraction $\Pr(\varphi^i|\alpha)$ it actually cares about for channel j). However, this is a very loose upper bound. We can tighten it significantly by ensuring that a bid's allocation does not exceed the supply that it actually cares about. That is, we add additional constraints of the form $x_\alpha^i \leq s(\varphi^i \wedge \alpha)$ for all bids i and channels α. This is still an overestimate because it does not account for interactions between multiple bids. However, empirically, this bound is quite close to an even tighter upper bound that we generate via constraint generation (see below).

Experimental results. Table 1 shows results from runs with parameters $MI = 0.01$ and $MI = 0.001$, averaged over 20 instances for each $\langle m, n \rangle$ pair. The table shows several key measures including the number of a-channels generated. The fraction of the upper bound on the optimal value obtained by the abstract LP when column generation terminates ("Frac UB") is also shown (giving us a lower bound on the quality of the abstract allocation relative to the true

[4] Lack of improvement does not imply allocation value is within *MI* of optimal, only that no *myopic* split that offers this improvement within our restricted space of splits: some sequence of splits could give more improvement. Even without this restriction (i.e., if splitting into arbitrary subsets is allowed), one can show that myopic splitting is insufficient under IP expressiveness. But for certain forms of LP expressiveness we can show that, unless the allocation is optimal, there exists a two-way split of some channel that improves value (in which case myopic splitting is sufficient).

	#	Frac		Runtime (sec)	
m	channels	UB	Improve	μ	range
			$MI = 0.01$		
10	12.0	0.893	0.447	12	$[4, 24]$
20	11.0	0.828	0.364	40	$[8, 74]$
30	10.2	0.841	0.380	75	$[35, 150]$
40	9.8	0.803	0.334	153	$[28, 556]$
50	10.0	0.816	0.396	212	$[23, 418]$
60	8.6	0.827	0.343	245	$[33, 470]$
70	8.3	0.824	0.304	314	$[26, 656]$
80	9.2	0.824	0.345	461	$[101, 940]$
90	8.6	0.806	0.333	566	$[75, 1211]$
100	9.3	0.804	0.344	811	$[203, 1438]$
			$MI = 0.001$		
10	32.4	0.965	0.515	53	$[10, 112]$
20	33.8	0.905	0.439	317	$[21, 758]$
30	27.1	0.899	0.438	538	$[112, 1384]$
40	28.6	0.871	0.399	1247	$[211, 4159]$
50	26.8	0.871	0.450	1543	$[153, 4027]$
60	22.7	0.877	0.392	1775	$[88, 4798]$
70	19.3	0.867	0.346	1959	$[66, 5878]$
80	24.2	0.873	0.393	3746	$[469, 8670]$
90	24.0	0.858	0.374	4956	$[807, 14534]$
100	25.7	0.853	0.392	6687	$[1677, 17047]$

Table 1: Average results for column generation with LP expressiveness m attributes, and $n = 10m$ bidders.

optimal allocation). An estimate of the improvement in the degree of optimality is shown ("Improve"). This is reported as the average of $(\textit{Final} - \textit{Initial})/\textit{UB}$, where *Final* is the final LP value, *Initial* is the LP value at the start of column generation (when a single abstract channel is used), and *UB* is the upper bound on the optimal value. Finally, the average and range of runtimes is presented.

We see that, with LP expressiveness, column generation can obtain a significant fraction of the upper bound value for problems in which it would be impossible to even enumerate the full unabstracted LP. Furthermore, the number of generated channels is comparable across all problem sizes tested. Setting a lower value for the minimum improvement parameter MI allows us to obtain a greater fraction of the upper bound, but with a fairly significant increase in run time. We note that, on average, much of the improvement is obtained early in the procedure. Fig. 1 shows the fraction of the upper bound obtained after a given number of channels has been generated, averaged over 20 instances, with 10 features, 100 bidders, and $MI = 0.001$. We obtain a high fraction of the upper bound from the first few channels generated, with additional channel splitting providing more modest improvement.

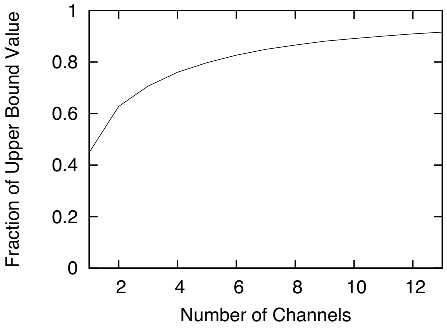

Figure 1: Fraction of upper bound vs number of channels for 10 attributes and 100 per-unit bidders.

	#	Frac		Runtime (sec)	
n	channels	UB	Improve	μ	range
10	6.6	0.847	0.248	41	$[5, 82]$
20	6.6	0.815	0.252	66	$[15, 129]$
30	7.0	0.769	0.264	91	$[14, 205]$
40	8.5	0.790	0.296	153	$[31, 282]$
50	8.8	0.823	0.325	188	$[39, 613]$
60	6.8	0.814	0.289	92	$[5, 325]$

Table 2: Average results for column generation with MIP expressiveness, $MI = 0.01$, 100 attributes, n bonus bidders, and $4n$ per-impression bidders.

MIP Expressiveness The second problem set adds all-or-nothing bonus bids to the per-impression bids above. Since these require binary variables, column generation on the LP relaxation only provides an approximation to the optimal abstract allocation. All problems have 100 attributes, n bonus bidders, and $4n$ per-impression bidders, with $n \in \{10, 20, \ldots, 60\}$. The preferences of per-impression bidders are as before. Each bonus bidder has φ^j and w^j chosen similarly; but its per-impression value is $v^j = 0$, and instead it pays b^j if it receives at least q^j impressions satisfying φ^j, and nothing otherwise. We set $q^j = \tau^j \sigma^j$ where $\tau^j \sim U[0.1, 1]$ is the fraction of the supply σ^j of φ^j in window w^j. We then set $b^j = \hat{b}^j q^j$ where $\hat{b}^j$ is chosen as v^j for a flat bidder, but then multiplied by a factor chosen from $U[1.1, 1.5]$. We also include a "market" bid as above.

Table 2 shows results with $MI = 0.01$, averaged over 20 instances for each n. Shown are the number of channels generated, the fraction of the upper bound (on the optimum) obtained when column generation terminates ("Frac UB"), the improvement over the fraction of the upper bound obtained before column generation ("Improve"), and the mean and range of runtimes. Although we use the LP relaxation to determine channel splits, the feasible allocation and the upper bound are computed by solving the corresponding MIP (discounted or not) on the set of channels produced.

Although column generation operates on a relaxation of the true MIP, our scoring function finds very good channel splits. Indeed, the performance with MIP expressiveness compares favorably to that with LP expressiveness. We emphasize that these campaign-level optimizations are run offline, and used to parametrize dispatch policies that are then implemented in real time. Thus the times reported here allow frequent, multiple optimizations (and reoptimization) of offline allocations (Boutilier *et al.* 2008).

Constraint Generation

The column generation approach converges to an optimal allocation with LP expressiveness (though we may not run it to optimality). It is not guaranteed to converge to optimality for MIPs since it is run on the LP relaxation at the root of the search tree. We develop a *constraint generation* algorithm that can be used to refine any abstract allocation, and will converge to optimality for MIPs, as well as for LPs.

The Constraint Generation Process

The optimization above, using the abstraction generated by our column generation process, assumes that any ad allocated to an a-channel α will be randomly dispatched to the component c-channels that make up α. This is reflected in the MIP (or LP) objective, where we replace the per-impression value v^i of bid i by $v^i_\alpha = v^i \Pr(\varphi^i|\alpha)$. With a well-crafted abstraction, this may produce an optimal allocation. However, if the number of a-channels is limited for computational reasons, the "pessimism" of random dispatch may leave revenue or efficiency on the table.

Alternatively, given an abstraction A, we can run an *optimistic MIP* assigning bids to a-channels *assuming* each impression to bid i satisfies its formula φ_i (i.e., do not discount the impressions by p^i_α). This optimistic assumption may not be valid—there may be no allocation of α to bids that permits feasible "packing" of their promised supply so that each i gets only φ^i-impressions. But we can *test* this assumption with a simple LP that determines whether there is enough supply to do so. Let $\dot{\mathbf{x}} = \{\dot{x}^i_\alpha\}$ be the solution of the optimistic MIP with a-channels $\{\alpha\}$. Let $W(\alpha) = \{i : \dot{x}^i_\alpha > 0\}$ denote the the "winners" of a-channel α. We solve the following LP for each α (with $x^i_c \geq 0$):

$$\begin{aligned} \min \quad & 1 \\ \text{s.t.} \quad & \sum_{c\in\alpha, c\models\varphi^i} x^i_c = \dot{x}^i_\alpha \qquad && \forall i \in W(\alpha) \\ & \sum_{i\in W(\alpha)} x^i_c \leq s(c) && \forall c \in \alpha. \end{aligned} \tag{2}$$

This LP determines a feasible allocation to the c-channels that constitute α, thus guaranteeing that every impression given to i satisfies its bid condition φ^i. The first set of constraints ensures there is enough φ^i supply for each bid i, while the second establishes that no c-channel is overallocated. If $LP(\alpha)$ is feasible for each α, then it provides an optimal dispatch policy that extracts the full objective value of the optimistic MIP.

If $LP(\alpha)$ is infeasible, then there must be some minimal set of constraints that are jointly infeasible. Let $S = S_a \cup S_s$ be such a minimal set, where S_a are constraints of the first type, and S_s are constraints of the second type. We can show that the MIP solution violates the inequality

$$\sum_{i\in S_a} x^i_\alpha \leq \sum_{c\in S_s} s(c).$$

We add this constraint to ensure that overallocation of the channels in S_s does not occur from bids in S_a. A tighter version of this constraint can be employed: we can add to the sum on the lefthand side any bid i all of whose relevant channels are included in S_s, i.e., any i s.t. $\{c \in \alpha : c \models \varphi^i\} \subseteq S_s$. At each iteration, sets S leading to violated constraints are identified for each a-channel and posted.[5] At each iteration, constraints are generated using a search procedure for identifying such sets S_s, and the MIP is resolved. This continues until feasibility is attained (in which case the optimistic objective value is actually obtained), or computational or time bounds are reached.

While $LP(\alpha)$ could require an exponential number of variables (i.e., the x^i_c corresponding to all c-channels $c \in \alpha$) and constraints, we use simple lossless channel abstraction (i.e., $\wedge_{i\in W(\alpha)} \pm \varphi^i$) to collapse this number. As such, the number of winners for each channel (and the interaction of their bids) determines the true complexity of the required LP solves. Even with lossless channel abstraction, the feasibility LP could require an exponential number of variables. In practice, we find that if $W(\alpha)$ is no greater than around 20, the size of the LP is reasonable (and *much* smaller than 2^{20}). If the MIP gives $W(\alpha) > 20$, we split channel α to minimize the maximum number of bids interested in a channel. Using this approach, we are able to generate LPs of reasonable size which solve very quickly (within a second).

The constraint generation algorithm can be used directly to solve the ad allocation MIP without relying on column generation. For example, it can be applied directly to the fully abstract MIP with a single a-channel ($\top$), or could be used to optimize w.r.t. *any* heuristically chosen abstraction.

Empirical Results

To evaluate the effectiveness of constraint generation we experiment with problems with bonus and per-impression bidders presented in the previous section. We first perform column generation using $MI = 0.01$, then extend the solution using constraint generation. To avoid generating an unreasonable number of constraints, we use a tolerance ϵ (set to 0.01) that permits MIP allocations to decrease by as much as ϵ, solving the following LP:

$$\begin{aligned} \min \quad & \epsilon \\ \text{s.t.} \quad & \sum_{c\in\alpha, c\models\varphi^i} x^i_c \leq \dot{x}^i_\alpha \qquad && \forall i \in W(\alpha) \\ & \sum_{c\in\alpha_j, c\models\varphi^i} x^i_c \geq \dot{x}^i_\alpha - \epsilon && \forall i \in W(\alpha) \\ & \sum_{i\in W(\alpha)} x^i_c \leq s(c) && \forall c \in \alpha. \end{aligned} \tag{3}$$

If constraint generation does not terminate within 600s., we stop the process and produce a feasible allocation that minimizes the maximum difference from the MIP allocation. Thus, when constraint generation terminates, the allocation may be suboptimal, but is guaranteed to be feasible.

When constraint generation is complete, we compute the value of the allocation based on the final feasible allocation generated by the LP (which might be different than that of the final MIP allocation, due to ϵ), but use the final (infeasible) MIP allocation as an upper bound on the true optimum value. This bound is close to, but somewhat tighter than the bound shown earlier.

Table 3 shows the results of our experiments: number of constraints generated; fraction of the upper bound on optimal value obtained by the MIP when constraint generation terminates ("Frac UB"); an estimate of *additional* improvement in the degree of optimality over the final column generation value ("Add'l improve"); and average and range of

[5] These can be identified using the facilities of standard solvers, such as the CPLEX IIS (irreducible inconsistent set) routine. We use our own special purpose algorithm to identify such sets.

	#	Frac	Add'l.	Runtime (sec)	
n	constraints	UB	improve	μ	range
10	221	0.954	0.104	154	$[14, 615]$
20	557	0.939	0.118	636	$[118, 1178]$
30	750	0.965	0.190	850	$[317, 1750]$
40	787	0.954	0.157	1434	$[648, 6609]$
50	721	0.967	0.139	1419	$[679, 6235]$
60	803	0.964	0.143	1029	$[635, 2269]$

Table 3: Average results from adopting the constraint generation phase following column generation, with IP expressiveness, $MI = 0.01$, 100 attributes, n bonus bidders, and $4n$ per-impression bidders.

constraint generation runtimes. The additional solve phase increases value to a high degree of optimality, finding solutions that are roughly within 94-97% of the upper bound (compared with 77-85% for column generation alone). Obtaining this improvement can be time consuming for larger problems. We emphasize, however, that abstraction generation is typically an offline process.

Other Uses of Constraint Generation

One bottleneck in the effective use of constraint generation is its poor scaling in the number of "winners." Specifically, if an a-channel, time-period pair has a large number of bids that are allocated to it in the initial abstract MIP solve, the procedure can generate hundreds of thousands of constraints, causing the MIP to slow down significantly and dominate runtime. The number of winners in the MIP can be used to suggest further channel refinements. The development of effective channel splitting heuristics that attempt to "separate" bids into different channels could make constraint generation much more effective. The quick identification of problematic a-channels during constraint generation is critical as well: whenever a channel is split, all constraints on the split channel must be discarded, and new ones must be generated on the new channels, further "wasting" computational effort. Thus problematic a-channels should be identified before significant constraint generation occurs.

Constraint generation can also be used selectively. The MIP can be solved by using the "optimistic" values on some channel-time pairs—requiring constraint generation to effectively carve up supply with those segments—while the random dispatch policy can be assumed in others (e.g., those where constraint generation cannot scale effectively). This offers a tractable means for improving on the abstract allocation problem without necessarily accounting for intelligent dispatch across the entire space.

Concluding Remarks

We developed a suite of techniques based on column and constraint generation that effectively tackle the channel explosion problem in the optimal allocation of display ads. Our techniques apply to both simple, current forms of expressiveness (e.g., simple budget constraints) and other, richer forms of campaign-level expressiveness that require the solution of large-scale integer programs (Boutilier *et al.* 2008; Parkes & Sandholm 2005). Our experiments demonstrate that high-quality allocations can be determined using very few abstract channels, indicating a desirable sensitivity of our methods to those distinctions that have the greatest impact on value (e.g., revenue or efficiency), and the ability to scale to problems with hundreds of attributes and bidders. Given the offline nature of the optimization problem we propose, our computational results suggest that our procedures can be run and rerun frequently to determine, say, (approximately) optimal allocations in stochastic models that require sampling (Boutilier *et al.* 2008).

Our method considers complex splits to generate a tractable number of channels. Though more sophisticated methods might further reduce the number of channels, our goal is not to minimize the number of channels per se, but to identify an abstraction with high value while maintaining LP/MIP tractability. Currently, channel split search dominates runtime: a focus of future research is accelerating this search, e.g., via heuristics for variable/literal ordering. Improvements to our constraint generation procedure are of interest as is the exploration of branch-and-price techniques. Finally, assessing the impact of approximate channel abstraction and optimization on incentives in ad markets would be of significant interest and value.

References

Abrams, Z.; Mendelevitch, O.; and Tomlin, J. 2007. Optimal delivery of sponsored search advertisements subject to budget constraints. In *ACM EC*, 272–278.

Benisch, M.; Sadeh, N.; and Sandholm, T. 2009. Methodology for designing reasonably expressive mechanisms with application to ad auctions. In *IJCAI*, 46–52.

Borgs, C.; Chayes, J.; Etesami, O.; Immorlica, N.; Jain, K.; and Mahdian, M. 2006. Bid optimization in online advertisement auctions. In *Workshop on Sponsored Search Auctions*.

Boutilier, C.; Parkes, D. C.; Sandholm, T.; and Walsh, W. E. 2008. Expressive banner ad auctions and model-based online optimization for clearing. In *AAAI*, 30–37.

Edelman, B.; Ostrovsky, M.; and Schwarz, M. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review* 97(1):242–259.

Feldmann, J.; Muthukrishnan, S.; Martin, P.; and Stein, C. 2007. Budget optimization in search-based advertising auctions. In *Workshop on Agent Mediated Electronic Commerce*.

Ghosh, A.; McAfee, P.; Papineni, K.; and Vassilvitskii, S. 2009. Bidding for representative allocations for display advertising. In *Fifth Workshop on Internet and Netowork Economics*, 208–219.

Lübbecke, M. E., and Desrosiers, J. 2005. Selected topics in column generation. *Operations Research* 53(6):1007–1023.

Parkes, D. C., and Sandholm, T. 2005. Optimize-and-dispatch architecture for expressive ad auctions. In *Workshop on Sponsored Search Auctions*.

Rusmevichientong, P., and Williamson, D. P. 2006. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In *ACM EC*, 260–269.

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