Scalable Segment Abstraction Method for Advertising Campaign Admission and Inventory Allocation Optimization

Fei Peng and **Tuomas Sandholm**
Optimized Markets, Inc.
Pittsburgh, PA, USA
{fei.peng, sandholm}@optimizedmarkets.com

Abstract

As publishers gather more information about their users, they can use that information to enable advertisers to create increasingly targeted campaigns. This enables better usage of advertising inventory. However, it also dramatically increases the complexity that the publisher faces when optimizing campaign admission decisions and inventory allocation to campaigns. We develop an optimal anytime algorithm for abstracting fine-grained audience segments into coarser abstract segments that are not too numerous for use in such optimization. Compared to the segment abstraction algorithm by Walsh *et al.* [2010] for the same problem, it yields two orders of magnitude improvement in run time and significant improvement in abstraction quality. These benefits hold both for guaranteed and non-guaranteed campaigns. The performance stems from three improvements: 1) a quadratic-time (as opposed to doubly exponential or heuristic) algorithm for finding an optimal split of an abstract segment, 2) a better scoring function for evaluating splits, and 3) splitting time lossily like any other targeting attribute (instead of losslessly segmenting time first).

1 Introduction

The business models of many Internet companies are based on individualized advertising. Display (aka. banner) ads and sponsored search ads are two prominent examples. When users visit a website or search for certain keywords, they are shown one or multiple ads based on the publisher's knowledge of them. Their browsing history, the device being used, and information gleaned from their social media accounts all feed into analysis that describes the socio-demographic characteristics of the user, and models that predict the user's behavior and response to certain ads. This allows the publishers to generate more revenue by dividing their *impressions*, or page views, into fine-grained *segments*, that is, classes of page views defined based on attributes (which can be based on user characteristics, time, URL, etc.).

Some companies sell impressions through auctions such as the generalized second price (e.g., Google) or Vickrey-Clarke-Groves auctions [Edelman *et al.*, 2007] (most notably Facebook). This allows each impression to be served only to advertisers interested in bidding on it, and leverages the publishers' knowledge about their users. However, the approach of selling one impression at a time cannot support many important types of constraints and preferences (aka. "expressiveness") that advertisers have—especially campaign-level constraints such as smoothness of delivery, reach, and campaign-level exclusivities.

In contrast, on the other end of sales approaches, manual sales of display ads support rich campaign execution constraints but very little targeting. The key planning problem for the publisher is to optimize

1. *campaign admission*: which campaign request to accept versus reject, and
2. *inventory allocation*: how to allocate heterogeneous inventory to accepted campaigns with different targeting and campaign-level constraints.

Typically, the publisher would like to optimize these so as to maximize revenue. The optimization problem behind these decisions becomes more complex with more campaigns, impressions, and increased targeting and campaign constraints.

With *non-guaranteed* campaigns, the admission decision can be left as a side effect of inventory allocation optimization: if no inventory is allocated to a campaign, it is implicitly rejected. With *guaranteed* campaigns, sometimes the admission decisions are made (e.g., manually) before the planning optimization, in which case the optimization only addresses part (2), but potentially with penalties for not fulfilling accepted campaigns.

The inventory allocation part of the planning problem has been considered for display advertisements [Abrams *et al.*, 2007; Alaei *et al.*, 2009; Yang *et al.*, 2010; Turner, 2012] and in-game advertisements [Turner *et al.*, 2011]. Expectation- and sample-based optimization approaches have been developed for stochastic supply and demand [Boutilier *et al.*, 2008]. Sample-based approaches have been proposed with near-optimal solution quality for both a linear program (LP) setting assuming randomly permuted queries [Devanur and Hayes, 2009], and a problem with a quadratic goal that includes a representativeness criterion [Ghosh *et al.*, 2009; Vee *et al.*, 2010; Bharadwaj *et al.*, 2012]. Techniques have also been proposed for planning under reach and frequency constraints [Hojjat *et al.*, 2014].

While forecasts on actual impressions can be used in the planning phase, the uncertain and changing user arrivals, among other factors, can make the realized impressions during the *dispatch* phase different from the forecast. A popular approach for handling this is the *optimize-and-dispatch* architecture [Parkes and Sandholm, 2005] where a planner plans and replans offline, and gives the plan to a dispatch engine (ad server) to execute as closely as possible in real time (typically with well less than 100 milliseconds to make a decision for an incoming impression). Control-theoretic and model-based bid-adjustment methods have also been proposed for configuring the dispatcher and adjusting the bid values real-time [Chen *et al.*, 2011].

We focus on the planning problem in this paper, and discuss briefly in the conclusions how our algorithm and results may be used in the dispatch phase. Our techniques can be used for the campaign admission optimization, the inventory allocation optimization, or for jointly optimizing both. Our work covers guaranteed and non-guaranteed campaigns, as well as a mixture of both. Our work tackles the same problem as Walsh *et al.* [2009; 2010], but with dramatically better performance via three technique improvements.

2 Segment Abstraction and Optimization

As publishers/advertisers gain access to more user information, the number of targeting attributes grows. The number of segments grows exponentially with the number of attributes. This creates insurmountable problems for the vanilla optimization problem of deciding which campaign requests to accept and how to allocate inventory to accepted campaigns.

For this problem, Walsh *et al.* [2009; 2010] introduced the idea of *segment abstraction* (which they originally called *channel abstraction*). The idea is to first algorithmically abstract the segments to larger abstract ones in a way that does not significantly compromise the objective (typically revenue). The problem is abstracted enough so that it can then be solved by an optimizer for the campaign admission and/or inventory allocation problem.

The approach has since then become popular in ad inventory allocation and campaign admission problems (e.g., [Turner *et al.*, 2011; Turner, 2012; Hojjat *et al.*, 2014]). It has also been extended to dynamic segmentation in *marketing* optimization [Lu and Boutilier, 2015].

Sometimes abstraction can be *lossless* (although typically one needs to abstract more than that to reach an optimization problem size that is practical to solve). As an example of lossless abstraction, if bid b targets women residing in California, as long as its ad is shown to a woman from California, it does not matter whether she has children or not. Therefore, we can combine concrete segments $\{$Woman, California, Children$\}$ and $\{$Woman, California, No-Children$\}$ into an abstract segment $\{$Woman, California$\}$. This is lossless: considering only the abstract segment preserves revenue optimality.

As in prior work, throughout this paper, we will assume that bids quantitatively express preferences for campaigns. Types of preferences include the following.

- **Budget**: maximum amount of money to spend over the entire campaign flight, or on a per-day basis;
- **Targeting**: a campaign may specify a number of attributes and a set of values for each that describe the audience it is trying to reach. We can also treat time as a attribute, and handle it the same way as the other attributes. Section 3.4 will discuss the handling of time in detail;
- **Impression threshold**: a campaign may have limits as to how many impressions it will buy, or at least how many it needs in order to pay;

While many such preferences can be modeled as an LP, some will require the solution of a Mixed Integer Linear Program (MIP). The size of the LP/MIP becomes prohibitive already at moderate numbers of attributes; this is the very motivation of segment abstraction. In our experiments, we will conduct tests both with LP expressiveness and with MIP expressiveness (specifically guaranteed campaigns, where campaign admission becomes a meaningful decision).

Optimization Model without Segment Abstraction

We define the set of attributes to be $\mathcal{F}$, and the domain of an attribute in $\mathcal{F}$ as a finite set of attribute values. A *concrete segment* is defined as an instance over the universe of attribute instantiations $dom(\mathcal{F})$.

Let C be the set of all possible concrete segments, and let $s(c)$ be the supply of impressions in segment $c \in C$. Let B be the set of bids, each of which can be described by its budget g^b, a bid value v^b for each impression, and a targeting criterion ψ^b, which is a logical proposition over $dom(\mathcal{F})$. For ease of presentation, let $v_c^b = v^b$ if bid b targets segment c, and 0 otherwise. Finally, let x_c^b be the number of impressions we assign to bid b from segment c. The optimization model is

$$\begin{aligned}
\text{(MC)}\ \ \text{maximize}_x\ & \sum_b \sum_c v_c^b x_c^b & \\
\text{s.t.}\ & \sum_b x_c^b \le s_c & c \in C \\
& \sum_c v_c^b x_c^b \le g^b & b \in B \\
& x_c^b \in \mathbb{Z}, 0 \le x_c^b \le s_c. &
\end{aligned}$$

The possible number of segments in set C grows exponentially with the number of attributes, creating the problem of segment explosion that renders the direct solution of this problem impossible. As pointed out in Walsh *et al.* [2010], a naive but lossless (i.e., revenue-maximal) way of abstracting concrete segments is to consider all sensible abstract segments of the form $\wedge_{b\in B} \pm \psi^b$. However, the number of those segments is still exponential in the number of bids.

3 Our Optimal Abstraction Algorithm

Our optimal anytime segment abstraction algorithm is for the same problem as the one tackled by Walsh *et al.* [2009; 2010]. As we will show, ours yields two orders of magnitude improvement in run time and significant improvement in abstraction quality. This performance gain stems from three improvements: 1) a quadratic-time (as opposed to doubly exponential or heuristic) algorithm for finding an optimal split of an abstract segment, 2) a better scoring function for evaluating splits, and 3) splitting time lossily like any other targeting attribute (instead of losslessly segmenting time first).

The segment abstraction starts with an initial set of abstract segments (for example, one channel with the entire set of impressions), and iteratively splits existing segments into finer-grained ones. Given the set of abstract segments A, we consider the following *master problem*:

$$\begin{aligned} \text{(M)}\ \ \text{maximize}_x\ & \sum_b \sum_a v_a^b x_a^b \\ \text{s.t.}\ & \sum_b x_a^b \le s_a && a \in A \quad (1) \\ & \sum_a v_a^b x_a^b \le g^b && b \in B \quad (2) \\ & x_a^b \in \mathbb{Z}, 0 \le x_a^b \le s_a. \end{aligned}$$

Following standard practice, we will treat variables x_a^b as continuous since the effects of fractionality are negligible when the number of impressions is large.

Although Model (M) treats all concrete segments inside an abstract segment uniformly, we need to properly handle the fact that not all impressions in an abstract segment can be used to satisfy a given campaign. This is captured in the value parameter v_a^b:

$$v_a^b = P\{\psi^b \wedge a | a\} \cdot v^b,$$

where $P\{\psi^b \wedge a|a\}$ is the probability that a randomly assigned impression from a will satisfy campaign b:

$$P\{\psi^b \wedge a|a\} = \frac{s(\psi^b \wedge a)}{s(a)}.$$

By using v_a^b in the model, we are essentially *discounting* the value of an assigned impression by the proportion of impressions in a that b targets. [1]

3.1 Splitting an Abstract Segment

By splitting some abstract segments, one may often be able to improve the solution quality. We present an approach akin to column generation [Barnhart *et al.*, 1998; Lübbecke and Desrosiers, 2005], but different in that we are not generating one column but rather generating two and removing one. We consider a split of, say $\alpha \in A$ that is part of the current coarse abstraction, into two segments β and $\bar{\beta}$, with which we associate variables $x_{\alpha\wedge\beta}^b$ and $x_{\alpha\wedge\bar{\beta}}^b$ for $b \in B$. Define dual variables π_a for constraints (1) and δ^b for constraints (2). The *reduced cost*, or the change in the objective function for bringing the value of one such variable from 0 to a tiny amount, is

$$\begin{aligned} rc(x_{\alpha\wedge\beta}^b) &= v_{\alpha\wedge\beta}^b(1-\delta^b) - \pi_\alpha \\ rc(x_{\alpha\wedge\bar{\beta}}^b) &= v_{\alpha\wedge\bar{\beta}}^b(1-\delta^b) - \pi_\alpha. \end{aligned}$$

We define the *score* of the split into β and $\bar{\beta}$ as:

$$\begin{aligned} score(\alpha, \beta, \bar{\beta}) = & \max_{b\in B}\{rc(x_{\alpha\wedge\beta}^b) \cdot s(\alpha \wedge \beta)\} \\ & + \max_{b\in B}\{rc(x_{\alpha\wedge\bar{\beta}}^b) \cdot s(\alpha \wedge \bar{\beta})\}. \quad (3) \end{aligned}$$

Theorem 1. *If an abstract segment α^* is split into β^* and $\bar{\beta}^*$ in each iteration, where*

$$(\alpha^*, \beta^*, \bar{\beta}^*) = \arg\max\{score(\alpha, \beta, \bar{\beta})\},$$

the abstraction algorithm will terminate in a finite number of iterations at an optimal solution.

Proof. Essentially, the scoring function (3) assumes that after the split, the entire supply of segment β will be assigned to the bid with the highest discounted (by a factor of $\Pr(\psi^b \wedge (\alpha \wedge \beta)|\alpha \wedge \beta)$) value of a β impression, and all of the supply of $\bar{\beta}$ assigned to the bid with the highest discounted value of a $\bar{\beta}$ impression. When campaigns have enough budgets to buy all of a segment, this produces exactly the objective improvement. However, most of the time, this score is an overestimate of the true score from the split. If the maximum score is negative, we already know that no more improving splits can be found. Because the score is an overestimate of the true score, using this score can unnecessarily split a segment without improving the objective, or prolong the solution process without recognizing that the solution is already optimal. However, since we can only split a segment a finite number of times, the algorithm will at worst split all abstract segments into concrete segments, which provides an optimal solution. □

This result may not seem attractive at first, since in the worst case we may end up solving Problem (MC) in addition to spending effort in the abstraction process. However, this algorithm can be stopped at any time and provide a solution to (MC), while solving or even writing down (MC) is not possible. In reality, the splitting process can be terminated as soon as solution quality is good enough, and, as we will show in the experiments, this scoring function works well in finding high-quality solutions quickly in practice. Note that Theorem 1 is also true under the scoring function in Walsh *et al.* [2010], although by incorporating the dual variables for budget, our scoring function (3) can identify better splits than the scoring function in Walsh *et al.* [2010], as the experiments will show.

3.2 Finding a Split with the Maximum Score

In this section, we describe a procedure for finding a split, given the current set of abstract segments A, that has the maximum score (3). As pointed out in Walsh *et al.* [2010], the possible number of splits for each segment α is $2^{k^{\mathcal{F}}}$, or doubly exponential in the number of attributes and the domain size of each attribute. Therefore, trying to find the best

[1] This leads to the correct estimation of objective value if a straightforward dispatcher is used that simply assigns impressions randomly according to the planned probabilities x_a^b. If a smarter dispatcher is used that determines before dispatching whether there is a fit between the impression and the bid targeting criteria, this model can underestimate the objective value.

split by enumerating all possibilities is unrealistic. Walsh *et al.* [2010] used heuristics for curtailing that search. We now show that one can find an optimal split in quadratic time in the number of bids!

Examining the scoring function, and letting $\bar{\delta}^b = 1 - \delta^b$, we have

$$
\begin{aligned}
& score(\alpha, \beta, \bar{\beta}) \\
= & \max_{b \in B} \left\{ \left(v^b_{\alpha\wedge\beta}\, \bar{\delta}^b - \pi_\alpha\right) s(\alpha \wedge \beta) \right\} \\
& + \max_{b \in B} \left\{ \left(v^b_{\alpha\wedge\bar{\beta}}\, \bar{\delta}^b - \pi_\alpha\right) s(\alpha \wedge \bar{\beta}) \right\} \\
= & \max_{b \in B} \left\{ v^b_{\alpha\wedge\beta}\, \bar{\delta}^b\, s(\alpha \wedge \beta) - \pi_\alpha\, s(\alpha) \Pr(\alpha \wedge \beta | \alpha) \right\} \\
& + \max_{b \in B} \left\{ v^b_{\alpha\wedge\bar{\beta}}\, \bar{\delta}^b\, s(\alpha \wedge \bar{\beta}) - \pi_\alpha\, s(\alpha) \Pr(\alpha \wedge \bar{\beta} | \alpha) \right\} \\
= & \max_{b \in B} \left\{ v^b_{\alpha\wedge\beta}\, \bar{\delta}^b\, s(\alpha \wedge \beta) \right\} + \max_{b \in B} \left\{ v^b_{\alpha\wedge\bar{\beta}}\, \bar{\delta}^b\, s(\alpha \wedge \bar{\beta}) \right\} \\
& - \pi_\alpha\, s(\alpha) \\
= & \max_{b \in B} \left\{ v^b \Pr(\psi^b | \alpha \wedge \beta)\, \bar{\delta}^b\, s(\alpha \wedge \beta) \right\} \\
& + \max_{b \in B} \left\{ v^b \Pr(\psi^b | \alpha \wedge \bar{\beta})\, \bar{\delta}^b\, s(\alpha \wedge \bar{\beta}) \right\} - \pi_\alpha\, s(\alpha) \\
= & \max_{b \in B} \left\{ v^b\, \bar{\delta}^b\, s\left(\psi^b \wedge (\alpha \wedge \beta)\right) \right\} \\
& + \max_{b \in B} \left\{ v^b\, \bar{\delta}^b\, s\left(\psi^b \wedge (\alpha \wedge \bar{\beta})\right) \right\} - \pi_\alpha\, s(\alpha).
\end{aligned}
$$

Theorem 2. *For any given abstract segment, a split with maximum score (3) can be found by evaluating* $\frac{|B|^2 - |B|}{2}$ *splits.*

Proof. Consider abstract segment α. Because we know that $\max(score(\alpha, \beta, \bar{\beta}))$ will happen for some pair of bids (b^*, b'^*), if we can find the split β that maximizes

$$v^b\, \bar{\delta}^b\, s\left(\psi^b \wedge (\alpha \wedge \beta)\right) + v^{b'}\, \bar{\delta}^{b'}\, s\left(\psi^{b'} \wedge (\alpha \wedge \bar{\beta})\right)$$

for each pair of bids b and b', the largest of all $\frac{|B|^2 - |B|}{2}$ such scores will be the maximum score ($\pi_\alpha s(\alpha)$ is a constant for a given α).

Consider Figure 1 as an illustrative example with two bidders b and b'. Bid b targets all concrete segments in regions 1 and 3, while b' targets 2 and 3. The rectangle represents the entire segment α. If region 3 is empty, the maximum-score split with regard to only b and b' is to include 1 in β, include 2 in $\bar{\beta}$, and arbitrarily assign the rest of the segment to either β or $\bar{\beta}$. On the other hand, when region 3 is not empty, the split that achieves the maximum score for b and b' should assign 1 to β, and 2 to $\bar{\beta}$ (since b pays 0 for any impression in 2 and b' pays 0 for any impression in 1). Moreover, 3 should be assigned to b only if $v^b \bar{\delta}^b > v^{b'} \bar{\delta}^{b'}$, and vice versa. Evaluating all such assignments will produce a maximum-score split. □

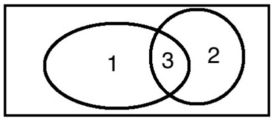

Figure 1: Illustration of the targeting criteria of two bids.

We describe in Algorithm 1 the pseudocode for finding the split with the maximum score for a segment α.

Algorithm 1 `MaxScore`(α)

```
Let $maxscore = 0$
for each $i \in \{1, \ldots, |B|\}$, let $b = b^i$ do
    for each $j \in \{i, \ldots, |B|\}$, let $b' = b^j$ do
        if $\psi^b \wedge \psi^{b'} = \emptyset$ then
            $score = v^b \delta^b s(\alpha \wedge \psi^b) + v^{b'} \delta^{b'} s(\alpha \backslash \psi^b)$
        else if $v^b \delta^b > v^{b'} \delta^{b'}$ then
            $score = v^b \delta^b s\left(\alpha \wedge \psi^b\right) + v^{b'} \delta^{b'} s\left(\alpha \backslash \psi^b\right)$
        else
            $score = v^{b'} \delta^{b'} s(\alpha \wedge \psi^{b'}) + v^b \delta^b s(\alpha \backslash \psi^{b'})$
        if $score > maxscore$ then
            $maxscore \leftarrow score$
Return $maxscore - \pi_\alpha s(\alpha)$
```

The same procedure can be applied to the scoring function proposed in Walsh *et al.* [2010].

3.3 Our Entire Abstraction Algorithm

Now we are ready to present our algorithm for generating the set of abstract segments that yields optimal revenue.

1. Initialize the set A with some initial abstract segments. For example, A can include only one segment with the entire set of impressions.
2. Solve the master problem to get the dual solution.
3. For each abstract segment in A, run `MaxScore`(α) to find the split with the maximum score.
4. If `MaxScore`(α) ≤ 0 for all $\alpha \in A$, declare optimality and terminate; otherwise split the segment with the highest score and go to Step 2.

3.4 Time As an Attribute

There are at least three possible ways that time could be handled in our segment abstraction approach:

1. No time abstraction. Time is first discretized as finely as needed into non-overlapping units, and the abstraction is only applied to other attributes.
2. First abstract the discretized time units into intervals while making sure that each campaign is indifferent to individual time units inside an interval. Then run the abstraction algorithm on other attributes. Compared to the first approach, this can reduce the size of the optimization problem when the number of bids is small, but tends to lead to the same time abstraction as the first approach when the number of bids is large because there tend to then be no two time units that no bid distinguishes among.

   Approaches 1 and 2 handle time in a *lossless* way.

3. The most versatile of these approaches is to handle time abstraction together with the abstraction of other attributes in the unified abstraction framework that we described. We let the algorithm determine the most beneficial splitting of time and/or other attributes. This is *lossy* time abstraction.

4 Experiments

For the experiments, we built an instance generator for different problem types and sizes, and configured it to behave identically to that of Walsh *et al.* [2010]. We then compared results on test cases that are the same size as theirs. We now describe the settings of the generator.

Consider a time horizon of 30 days, where a day is the finest unit of time targeting allowed. The total supply of impressions is 1,000,000 per day. An impression has, independently for each attribute $i \in \mathcal{F}$, value f_i^1 with probability $\Pr(f_i^1) = U[0,1]$, and $\Pr(f_i^2) = 1 - \Pr(f_i^1)$. To model commonality in bids' targeting, the popularity of attribute $i \in \mathcal{F}$ follows a Zipf distribution $P_i = (1/i)/(\sum_{\ell=1}^{|\mathcal{F}|} 1/\ell)$. Each bid cares about a set of attributes I_b, where $|I_b| \sim U[0,10]$, with each attribute sampled from the Zipf distribution without replacement. A bid has uniform preferences $U[0,1]$ over values for each attribute it targets. The value bid b places on each matching impression is $v_b = \hat{v}_b(1 + 10\sum_{i \in I_b} P_i)$, where $\hat{v}_b$ is a base value sampled from $U[0.1,1]$. In other words, highly targeted campaigns pay more for each impression, as is typical in practice. Moreover, we sample two time points $t_1, t_2 \sim U[-10,40]$ for each bid, use $[\min(t_1,t_2), \max(t_1,t_2)]$ as the bid's time window, and truncate it to $[1,30]$, our time horizon of interest. Bid b's budget is set to cover a fraction $\tau_b \sim U[0.1,1]$ of the matching impressions, σ_b, over its time window: $g^b = \tau_b\, \sigma_b\, v_b$. Finally, to capture opportunity cost of impressions, which may be the value of an impression on an external advertising network, an additional bid is introduced with per-impression value 0.1, unlimited budget and no targeting.

We consider two groups of problem instances, replicating those in Walsh *et al.* The *non-guaranteed* group contains campaigns that pay their per-impression value v_b for every impression delivered. The *guaranteed* group contains both per-impression bids, and bids that pay g^b only when they receive their total requested number of impressions (and 0 otherwise). A guaranteed campaign's value and budget are scaled up by a factor $\sim U[1.1, 1.5]$ to reflect their more stringent requirements. For the non-guaranteed group, we run 5 problem sizes with number of attributes $m \in \{20, 40, 60, 80, 100\}$, and number of bidders $n = 10m$. For the guaranteed group, we test 6 sizes with 100 attributes, $n \in \{10, 20, 30, 40, 50, 60\}$ guaranteed campaigns and $4n$ non-guaranteed campaigns. The optimization problem for the first (non-guaranteed) group is an LP. We formulate a MIP to capture the guaranteed campaigns in the second problem group, where a binary variable is associated with each guaranteed campaign to represent whether the required number of impressions is allocated to the campaign or not, but the segment abstraction is based on the relaxation of this MIP to an LP.

The solution to our optimization problem will not only include allocation decisions but also admission decisions. For the non-guaranteed group, campaigns that receive zero impressions are rejected. For the guaranteed group, the value of the binary decision determines whether the campaign is accepted or not.

The quality of the solution is evaluated by comparing to the same upper bound as the one used by Walsh *et al.*: replace all v_a^b in problem (M) by v^b, and impose an upper bound on x_a^b so that each bidder can get at most its matching number of impressions in each segment a. Since bids are paying for the un-discounted value of each impression, this is an upper bound on the optimal revenue. The upper bound problem and the MIP (for guaranteed campaigns) are only solved to obtain performance metrics in different iterations, and will not count towards the time to compute an abstraction.

The experiments were conducted on an Amazon Web Services instance of type r3.xlarge with 4 cores and 30.5GB of memory. Our implementation was single threaded. The results below are averaged over 20 instances for each problem size.

4.1 Time Abstraction Techniques

In Figure 2, we compare the better two of our three time-handling techniques: 2 and 3. We ran each until it reached 100 segments. The advantage of letting the algorithm determine the best time segmentation (lossy time abstraction) is clear: while both versions would eventually converge to the optimal solution (as indicated by the quality relative to upper bound) in a finite number of steps, the lossy time abstraction achieves a much better solution with any given number of abstract segments. This advantage is consistent across problem sizes (each of the solid curves corresponds to a different

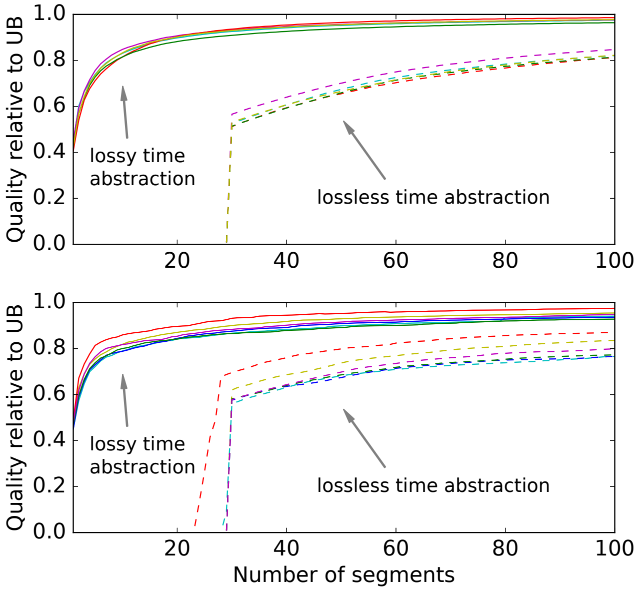

Figure 2: Lossy time abstraction (solid) vs. lossless time abstraction (dashed) for the non-guaranteed (top) and guaranteed (bottom) group.

problem size n, as does each of the dashed curves; within each curve cluster, problem size increases from high to low). The advantage applies to both groups: non-guaranteed and guaranteed.

4.2 Comparing Abstraction Algorithms

Next, we compare our algorithm to the one of Walsh *et al.*. In Figure 3, we mark with stars the quality of the solutions achieved in their paper (under setting $MI = 0.01$, the default setting there for termination), and run our algorithm until the same termination time for the corresponding problem size. At termination, the solutions generated by our algorithm with the new scoring function (solid lines) significantly outperform those reported in Walsh *et al.* for both the non-guaranteed and guaranteed groups. Again, same-colored lines and markers correspond to the same problem size n in each sub-figure, and problem size increases from high to low within each curve cluster. Moreover, in reaching the same solution quality, our algorithm achieves a speedup of up to 45x and 142x for the non-guaranteed and guaranteed groups, respectively.

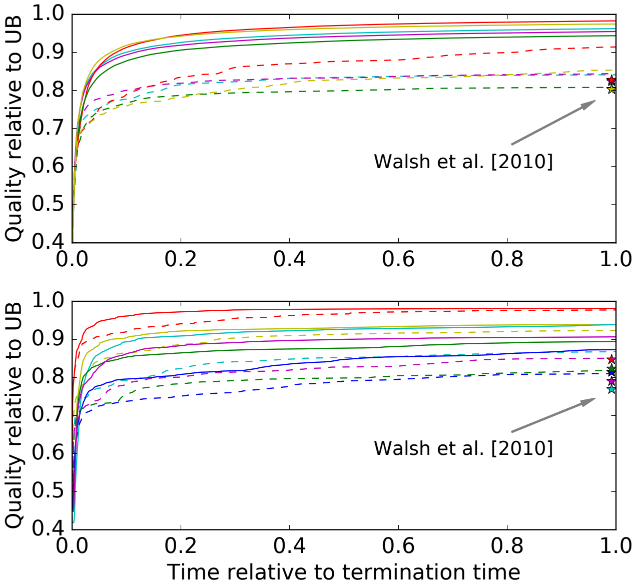

Figure 3: Revenue from our new abstraction algorithm (solid lines) compared to that of Walsh *et al.* (points represented by stars on the right). Also shown are our lossy abstraction solutions with old (dashed) and new (solid) scoring functions. The x-axis is time relative to the time reported by Walsh *et al.* for each of the problem sizes separately. The top figure is for the non-guaranteed group and the bottom figure for the guaranteed group.

Finally, we compare the effects of our new scoring function (3) versus the one used in Walsh *et al.*. We run our new fast lossy abstraction algorithm with these two scoring methods on same instances until the termination time, and track the progress in terms of percentage of upper bound reached. As can be seen in Figure 3, the new scoring function offers significant revenue gains, especially during the early stages. (Note further that the complexity of the master problem and of the split search are roughly equal between the two scoring functions, so the number of segments should be roughly proportional to time.)

5 Conclusions and Future Research

We developed an optimal anytime algorithm for segment abstraction. Compared to that of Walsh *et al.* [2010], it yields two orders of magnitude speed improvement and significant improvement in abstraction quality. These benefits hold both for guaranteed and non-guaranteed campaigns. The performance gain stems from three improvements: 1) a quadratic-time (as opposed to doubly exponential or heuristic) algorithm for finding an optimal split of an abstract segment, 2) a better scoring function for evaluating splits, and 3) splitting time lossily like any other targeting attribute (instead of losslessly segmenting time first). The algorithm makes rapid progress initially, and tapers off as it spends more time splitting segments more finely, especially for larger problem instances in the guaranteed group.

It is easy to parallelize our split search to speed up the execution of the algorithm. Further improvements can be obtained by adding a cut generation routine described in Walsh *et al.* [2010], which can be applied to the column generation phase both in that paper and here. In that setting, the best timing for switching from the column generation phase to the cut generation phase still remains open.

At the end of our segment abstraction process, the final set of segments, along with the planned probabilities x_a^b, are given to a dispatcher, which then serves impressions to individual campaigns according to these probabilities in real time. To prevent wasteful assignment of impressions, a smart dispatcher can determine, before dispatching, whether there is a fit between the impression and the bid targeting criteria. In addition, the dispatcher can try to accelerate campaigns that are falling behind plan in delivery, and vice versa (e.g., [Parkes and Sandholm, 2005; Chen *et al.*, 2011; Bharadwaj *et al.*, 2012]). However, because the time for a dispatcher to make a decision is extremely short, it cannot perform tasks for making sophisticated bid adjustments. Therefore, if the actual stream of impressions deviates significantly from forecast, periodically re-optimizing in light of delivered impressions may be warranted.

A number of issues remain to be investigated further. Handling other types of constraints besides budget, supply, demand, and targeting is of interest. Also, following the same instance generator setup as Walsh *et al.* [2010], we assumed that attributes are independent of each other. Having a more realistic, but still tractable attribute model can be useful. Moreover, how should the algorithm be intelligently adapted to settings where inventory forecasts can be inaccurate, and where bids may arrive dynamically? Finally, it would be interesting to take the approach to other large allocation problems.

References

[Abrams *et al.*, 2007] Zoe Abrams, Ofer Mendelevitch, and John Tomlin. Optimal delivery of sponsored search advertisements subject to budget constraints. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 272–278, 2007.

[Alaei *et al.*, 2009] Saeed Alaei, Esteban Arcaute, Samir Khuller, Wenjing Ma, Azarakhsh Malekian, and John Tomlin. Online allocation of display advertisements subject to advanced sales contracts. In *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, pages 69–77. ACM, 2009.

[Barnhart *et al.*, 1998] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[Bharadwaj *et al.*, 2012] Vijay Bharadwaj, Peiji Chen, Wenjing Ma, Chandrashekhar Nagarajan, John Tomlin, Sergei Vassilvitskii, Erik Vee, and Jian Yang. Shale: an efficient algorithm for allocation of guaranteed display advertising. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1195–1203, 2012.

[Boutilier *et al.*, 2008] Craig Boutilier, David Parkes, Tuomas Sandholm, and William Walsh. Expressive banner ad auctions and model-based online optimization for clearing. In *AAAI*, 2008.

[Chen *et al.*, 2011] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1307–1315, 2011.

[Devanur and Hayes, 2009] N. Devanur and T. Hayes. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *Proceedings of ACM Conference on Electronic Commerce (EC)*, 2009.

[Edelman *et al.*, 2007] Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *The American Economic Review*, 97(1):242–259, March 2007.

[Ghosh *et al.*, 2009] A. Ghosh, P. McAfee, K. Papineni, and S. Vassilvitskii. Bidding for representative allocations for display. In *Proceedings of the Workshop on Internet Economics (WINE)*, pages 208–219, 2009.

[Hojjat *et al.*, 2014] Ali Hojjat, John Turner, Suleyman Cetintas, and Jian Yang. Delivering guaranteed display ads under reach and frequency requirements. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

[Lu and Boutilier, 2015] Tyler Lu and Craig Boutilier. Value-directed compression of large-scale assignment problems. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[Lübbecke and Desrosiers, 2005] Marco E Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[Parkes and Sandholm, 2005] David Parkes and Tuomas Sandholm. Optimize-and-dispatch architecture for expressive ad auctions. In *First Workshop on Sponsored Search Auctions, at the ACM Conference on Electronic Commerce*, Vancouver, BC, Canada, June 2005.

[Turner *et al.*, 2011] John Turner, Alan Scheller-Wolf, and Sridhar Tayur. Scheduling of dynamic in-game advertising. *Operations Research*, 59(1):1–16, 2011.

[Turner, 2012] John Turner. The planning of guaranteed targeted display advertising. *Operations Research*, 60:18–33, 2012.

[Vee *et al.*, 2010] E. Vee, S. Vassilvitskii, and J. Shanmugasundaram. Optimal online assignment with forecasts. In *Proceedings of ACM Conference on Electronic Commerce (EC)*, 2010.

[Walsh *et al.*, 2009] William Walsh, Craig Boutilier, Tuomas Sandholm, Robert Shields, George Nemhauser, and David Parkes. Automated channel abstraction for advertising auctions. In *Proceedings of the Ad Auctions Workshop*, 2009.

[Walsh *et al.*, 2010] William Walsh, Craig Boutilier, Tuomas Sandholm, Robert Shields, George Nemhauser, and David Parkes. Automated channel abstraction for advertising auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2010.

[Yang *et al.*, 2010] Jian Yang, Erik Vee, Sergei Vassilvitskii, John Tomlin, Jayavel Shanmugasundaram, Tasos Anastasakos, and Oliver Kennedy. Inventory allocation for online graphical display advertising. *arXiv preprint arXiv:1008.3551*, 2010.