Traffic Shaping to Optimize Ad Delivery

DEEPAYAN CHAKRABARTI, McCombs School of Business, University of Texas, Austin
and ERIK VEE, Google Inc.

Web publishers must balance two objectives: how to keep users engaged by directing them to relevant content, and how to properly monetize this user traffic. The standard approach is to solve each problem in isolation, e.g., by displaying content that is tailored to the user's interests so as to maximize clickthrough rates (CTR), and also by building a standalone ad serving system that displays ads depending on the user's characteristics, the article being viewed by the user, and advertiser-specified constraints. However, showing the user only the articles with highest expected CTR precludes the display of some ads; if the publisher had previously guaranteed the delivery of a certain volume of impressions to such ads, then *underdelivery* penalties might have to be paid. We propose a joint optimization of article selection and ad serving that minimizes underdelivery by *shaping* some of the incoming traffic to pages where underperforming ads can be displayed, while incurring only minor drops in CTR. In addition to formulating the problem, we design an online optimization algorithm that can find the optimal traffic shaping probabilities for each new user using only a cache of one number per ad contract. Experiments on a large real-world ad serving web portal demonstrate significant advantages over the standalone approach: a threefold reduction in underdelivery with only 10% drop in CTR, or a 2.6-fold reduction with a 4% CTR drop, and similar results over a wide range.

Categories and Subject Descriptors: H.1.0 [**Models and Principles**]: General

General Terms: Economics, Algorithms, Performance

Additional Key Words and Phrases: Ad serving, Online reconstruction, Underdelivery

ACM Reference Format: B_ℓ Deepayan Chakrabarti and Erik Vee, 2015. Traffic shaping to optimize ad delivery. *ACM Trans. Embedd. Comput. Syst.* 9, 4, Article 39 (March 2010), 20 pages. B_ℓDOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Online display advertising is the fastest-growing segment of global advertising, growing by 18.9% a year [Zenith Optimedia 2011]. Each display of an ad to a user is called an *impression*, and in display advertising, advertisers pay for impressions; user actions such as clicking on ads are not necessary. Display advertising campaigns are often "targeted": Advertisers specify that their ad can only be shown to users visiting webpages where the (user, webpage) pair matches certain desired characteristics (e.g., the user is a young male who has previously shown interest in sports, and the webpage is about soccer). Targeting can even be more fine-grained, e.g., the ad can only be shown on certain positions on the webpage (called *ad slots*, such as "north", "lrec", etc.). We focus on *guaranteed* display advertising where the advertiser is guaranteed a certain number

Part of the work was done while the authors were at Yahoo! Research and Facebook.
Authors' addresses: D. Chakrabarti, IROM, McCombs School of Business, University of Texas, Austin (deepay@utexas.edu); E. Vee, Google Inc. (erikvee@google.com).

of qualifying impressions on the ad within a predetermined ad lifetime, and the price per impression is fixed.

Such guarantees can only be made if the publisher of the website can accurately forecast user traffic for the lifetime of the ad. In addition, the fraction of this traffic that matches the desired targeting attributes must also be forecast. Unfortunately, unbiased forecasting is very difficult in both cases, and the high variances associated with the forecasts only compounds the problem. Hence, the publisher must be conservative when guaranteeing impressions, or run the risk of being liable for severe underdelivery penalties. Even if the eventual supply of impressions exceeds forecasts, those can only be sold on the spot market for *non-guaranteed* display advertising, where prices are much lower than in the guaranteed marketplace. Hence, the publisher is unable to fully utilize the incoming user traffic.

At the core of the problem lies the following constraint: the publisher has no control over the volume or characteristics of the user traffic visiting her website. However, this constraint *can* be relaxed if we also model the flow of users from webpage to webpage. For instance, if the publisher is able to influence a user u arriving on webpage w_1 to visit another webpage w_2 such that the pair (u, w_2) matches the targeting attributes of an underdelivering ad[1], and this ad can be displayed on the ad slots available on w_2, then the publisher has essentially created a revenue-generating impression. Such *traffic shaping* can dynamically alter the distribution of user traffic across webpages to reduce underdelivery, allowing the publisher to guarantee larger impression volumes to advertisers. However, any attempt to influence users carries with it the risk of degrading the user experience. Thus, traffic shaping must couple the dual objectives of reducing underdelivery as much as possible while allowing only insignificant drops in user satisfaction metrics.

To make these ideas more precise, we focus on a case study involving a large web portal. A user arriving at the portal homepage is shown a set of clickable article abstracts (e.g., the day's top stories displayed on www.yahoo.com). Clicking on an abstract takes the user to the full article webpage, and ads can be displayed on any ad slots available on the article. We assume that each article has an arbitrary number of ad slots that is known in advance. Users can also visit these article webpages directly (say, via a search engine). The number of impressions for an ad is then the aggregate number of user visits on webpages where this ad is displayed.

Measuring the user's "satisfaction" is a thornier issue. Several short-term metrics are available, such as the clickthrough rate (CTR) of users on the displayed article summaries, or the dwell-time on the article landing page. Satisfaction over the long run can be measured by the total time spent by users on the portal, the number of repeat visits, the number of active user accounts, etc. Ideally, one would like to optimize the long-term metrics, but their understanding is still in a nascent stage. Among the short-term metrics, the CTR is broadly used, either by itself or as a part of some other overall metric. It also offers ease of analysis and measurement, due to which we use it as the sole metric of satisfaction in our study.

Now, to optimize for user satisfaction (as measured by CTR), the obvious policy of the portal's publisher is to display articles that are most likely to be clicked by the user. If the user's attributes are known (e.g., the user might have been identified by cookie, or she could have logged in to the portal), then the portal can dynamically select, from an available pool of articles, the subset of articles that is best suited to the user's interests. However, suppose an ad from a sportswear company is currently underdelivering. The advertiser has specified that this ad can only be shown to young females who visit a

[1]An underdelivering ad is one for which the forecasted impressions over the remaining lifetime of the ad will be insufficient to meet its guarantee.

sports-related webpage. Then, when a matching user visits the portal, the publisher could preferentially display sports-related article abstracts to this particular user[2]. This might lead to a loss in CTR since the displayed articles might have lower predicted CTR as compared to other articles available in the pool, but it increases the publisher's utility — the number of impressions delivered to the underdelivering ad (if the user does click on the sports-related articles)[3].

The goal of traffic shaping would be to generate as many such impressions on underdelivering ads as possible, while accepting a bounded loss in CTR.

A seemingly simple solution would be to preferentially display those article summaries which, if clicked, would enable the display of the most underperforming ads. Improved variants of this strategy have been proposed in prior work as well [Agarwal et al. 2011]. However, these are reasonable only when the volume of shaped traffic is relatively insignificant; otherwise, the very act of traffic shaping may cause impression forecasts for ads to be grossly inaccurate, thereby impairing the determination of underdelivering ads. Thus, while such strategies are useful in some scenarios, they fall short when a general solution is required.

Instead, in this paper, we seek a unified mathematical formulation of the traffic shaping problem. It should accept three inputs: (a) forecasts of traffic *arriving at the portal* (and hence, independent of any previous traffic shaping), (b) a black box that predicts the expected CTR when a user is shown an article abstract, and (c) another black box that uses the targeting attributes of all available ads to determine which ads can be shown on each ad slot ℓ of a given article page p if it is visited by user u. We implicitly assume that enough characteristics of future articles are known as to enable computation of their expected CTR and the number of ad slots on a webpage displaying the full article[4]. Now, whenever a new user arrives at the portal's website, we must optimally select one article abstract to show to the user. We note that the problem of picking multiple articles is the same as the single-article case, except for the need to adjust expected CTR depending on article position (as is common for, say, search result ranking), so we focus on the single-article problem for ease of exposition. Business reasons might also constrain traffic shaping by requiring that certain articles are shown to some minimum (or maximum) fraction of users. For instance, the publisher could choose to show to 90% of users the article with the maximum expected CTR, and leave only 10% for traffic shaping; this would also lower-bound the CTR after traffic shaping. As another example, the publisher could insist that every article receives at least 5% of the traffic, perhaps to get better CTR estimates. Finally, since runtime speed is critical, the algorithm should only require access to the set of matching ads and possibly a small cache; loading into memory the entire forecasted traffic is out of the question. Thus, we need to not only formalize the optimization problem, but also make it efficiently solvable. This dual requirement is what makes the problem difficult.

Our contributions are as follows.

(1) *Formulation:* We formulate traffic shaping as an optimization problem. Our framework is very generic, and incorporates domain knowledge and constraints that are of practical importance: forecasts of user traffic, differences in the ad slots avail-

[2]Note that the user is merely browsing and has not expressed any *intent*, e.g. by issuing a search query. Thus, article selection involves suggesting articles that may be of interest to the user based solely on his/her prior interactions with the website. In particular, search results are *not* biased.
[3]Note that users voluntarily click on links leading them to the webpage containing the full article. So, although it is possible to make an argument that clicks due to traffic-shaping give the publisher or advertiser less utility than un-shaped clicks, such an effect – if any – should be minor compared to other considerations.
[4]In practice, only estimates will be available, but these are good enough.

able on various article pages, constraints on the displayable ads depending on the user and article attributes, minimum (maximum) probabilities for displaying articles, etc.

(2) *Optimality:* We present an algorithm that, given the attributes of a user arriving at the portal's webpage, assigns probabilities to the various available article abstracts. One of these article abstracts is picked from this distribution and displayed to the user. We prove that this strategy is optimal, assuming the correctness of the forecasts.

(3) *Online reconstruction:* Our algorithm pre-computes a small cache that stores one number per ad. At runtime, for every user arriving at the portal, only the set of matching ads and their cached values are needed by the algorithm. Thus, the correct solution of the optimization problem is reconstructed for each new user from an online cache, without having to process the entire data every time.

(4) *Empirical results:* On a large real-world dataset of historical logs from a web portal, the algorithm yielded a threefold reduction in underdelivery with respect to the standalone approach that selects articles and serves ads independently. The drop in CTR was only 10% and can be varied according to the publisher's preferences; for example, a 2.6-fold reduction in underdelivery can be achieved for only a 4% CTR loss. This clearly demonstrates the usefulness of traffic shaping and the flexibility of our formulation.

The paper is organized is follows. In Section 2, we propose our formulation of traffic shaping as a constrained optimization problem on a graph linking users, articles, ad slots, and ad contracts. We also present our online reconstruction algorithm that finds the optimal parameters for traffic shaping for a new user arriving at the portal. Empirical evidence of the usefulness of our traffic shaping algorithm is presented in Section 3. In Section 4, we generalize our approach and show how similar methods can be applied to a broad class of problems. We review related prior work in Section 5, and conclude in Section 6.

2. PROPOSED METHOD

We shall split the description of our work into three parts. First, we will discuss the proposed model of traffic shaping. This includes forecasting of user traffic, and interlinking users with articles, the ad slots on article pages, and the ad contracts themselves to create a *traffic shaping graph*. Then, we present our formulation of the traffic shaping problem as an optimization problem based on this graph. This will provide the mathematical formalization of the model, and capture the various constraints stemming from practical concerns. Finally, we will present our online reconstruction algorithm that computes a probability distribution over available articles based on the attributes of the user arriving at the portal. The primary goal here is to limit the cache size, and our proposed algorithm only needs to store only one number per ad in its cache. This allows it to scale to millions of ads, satisfying the exacting requirements of even the largest ad-serving systems.

2.1. Model

The overall traffic shaping problem is shown in Figure 1. A user k arrives at the portal webpage, and the publisher can choose to show her one article abstract out of a set of available abstracts. The displayed article abstract is clicked by the user with some click-through rate (CTR) that depends on features of the (user, article) pair. If clicked, the user arrives at the article page i, which has multiple ad slots ℓ; different articles may have different slots available. For each (user, article, slot) triple, there can be several matching ads j. Note that the set of matching contracts depends on user and article attributes as well as the ad slot, so the graph will be different for each user. Our goal is to compute the *traffic shaping probabilities* w_{ki} (denoting the probability

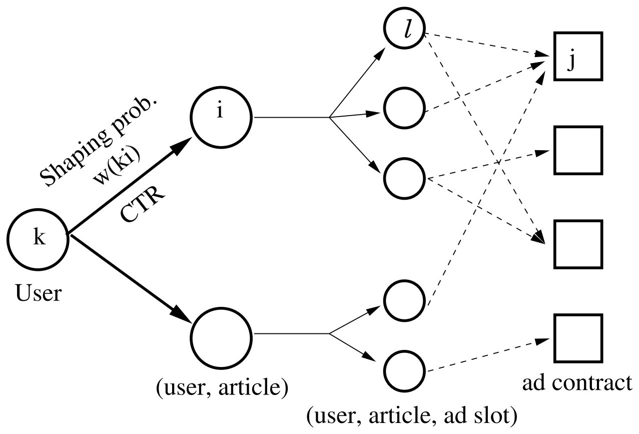


Fig. 1. Overall traffic shaping diagram

that user k is shown article page i) such that underdelivery of ads will be minimized, subject to constraints on these probabilities.

The model has two basic ingredients: the forecasts, and the graph linking the users, articles, and ads. Before describing these two aspects, we discuss why seemingly simpler methods of underdelivery estimation are inapplicable.

DIFFICULTY OF UNDERDELIVERY ESTIMATION. Traffic shaping algorithms would be simple and obvious if good underdelivery estimates were available for all ad contracts. Several apparently simple solutions present themselves, but each has significant flaws, as we discuss next.

The simplest idea is to estimate the future underdelivery of an ad from historical underdeliveries of "similar" ads. However, even if a reasonable similarity function could be defined, an ad underdelivers if not enough users with the right targeting attributes arrive at the website, or if such user impressions are cannibalized by other similar ads. Thus, such estimation is extremely sensitive to variations in the user traffic (by both type and volume) and the distribution and volume of ads in the entire system. In fact, such historical estimates are much more volatile than the original forecast problem which led to the overbooking in the first place, and generally too imprecise to be useful.

If an ad's underdelivery cannot be estimated, an alternative would be to forecast the total volume of user impressions to whom the ad could be shown. However, these impressions must be the (user, article, ad slot) triples mentioned above, and their volume is clearly dependent on traffic shaping itself. Forecasting numbers that are affected *by* traffic shaping, *for* the purposes of traffic shaping, via simple non-iterative methods appears difficult.

Finally, it might be possible to estimate underdelivery if the traffic shaping algorithm was very simple. However, our goal is to present a formal framework for traffic shaping and find the optimal solution. In Section 3, we do design an alternative greedy traffic shaping competitor; our proposed algorithm reduces underdelivery by over 61% as compared to it, while retaining the advantages of fast run-time execution thanks to our online reconstruction algorithm detailed later.

FORECASTS. Clearly, the probabilities w_{ki} must depend on the *other* users who visit the website in the future, since their ad views would affect the determination of underdelivering ads. If we could directly forecast the (user, article, ad slot) impressions that can be served to each ad, computing underdelivery would be easy; however, these future impression volumes depend on traffic shaping itself. To avoid such circularity, we need to forecast the expected number of users who will visit the portal *homepage* during the lifetime of the currently guaranteed ads, or will arrive directly at the article pages from some external site (say, a search engine).

In fact, we need forecasts of not only the *volume* of traffic but also the *distribution* of user attributes in it. However, there are many user attributes of interest to advertisers (e.g., age, gender, interests, geographical location, etc.), each of which can have many different values. Hence, a full specification of the attribute distribution can be very difficult to forecast and cumbersome to use.

An appealing alternative, described previously in [Vee et al. 2010], is to forecast *samples* of individual impressions instead of the entire probability distribution. Briefly, one creates an unbiased sample of future impressions and attaches, to each sample, a weight representing how likely it is to appear. Such a sample can be created from historical logs, e.g., by picking impressions from March, 2011 as samples for March, 2012; adjustments could also be made for any observed trends, such as changes in user demographics. An appropriate sample size can be chosen to trade off accuracy w.r.t. the full distribution against costs for computing time and storage space. The particular process employed in building this sample is orthogonal to our work, and any standard technique from the forecasting literature can be applied. Multiplying these sample weights by the forecasted traffic volume, we get a forecast of impressions that has both the expected volume and expected distribution of user attributes. We assume in our work that such a forecasted sample is made available as input to our algorithm.

One further complication is that new ads will enter the system over the forecasted time period, and they must be accounted for. We adjust for this effect by forecasting new ads as well, in a manner similar to that for forecasting sample impressions. Finally, we assume that even though the content of future articles is unknown, there will always be articles with the same *attributes* as those available currently. Thus, current articles can be used as proxies for all future articles.

LINKING USERS, ARTICLES, AND ADS. The forecasts give unbiased estimates of the future states of users, articles, and ads. These must be linked together to form the traffic shaping graph. The graph has four layers, successively connecting users to articles, then to ad slots on those articles, and finally the ads themselves. Figure 2 shows an example, with three forecasted users (A, B, and C) and two articles (white and black). Users A and B arrive at the portal homepage, while C bypasses the portal's homepage and arrives directly at one of the article pages. Clearly, only the traffic for A and B can be shaped, but C must be included since it affects the total forecasted traffic.

We reiterate a few points about this graph: (1) Articles can have different numbers of ad slots (one for the white article, three for the black article); (2) each node in the second and third layers includes a specific user attribute in its definition and can be traced back to a unique user node from the first layer, so there are no cross-connections in the first three layers, i.e., given ℓ, the corresponding k and i are fixed; (3) two users can be shown the same article and yet have different matching ads, since ads can target different user attributes; and (4) it is possible for some (user, article, ad slot) triples to have no matching ads. Next, we formalize the traffic shaping problem.

2.2. Mathematical Formulation

We first present some notation. We index users by k, (user, article) pairs by i, (user, article, ad slot) triples by ℓ, and ad contracts by j. For ease of exposition, we shall

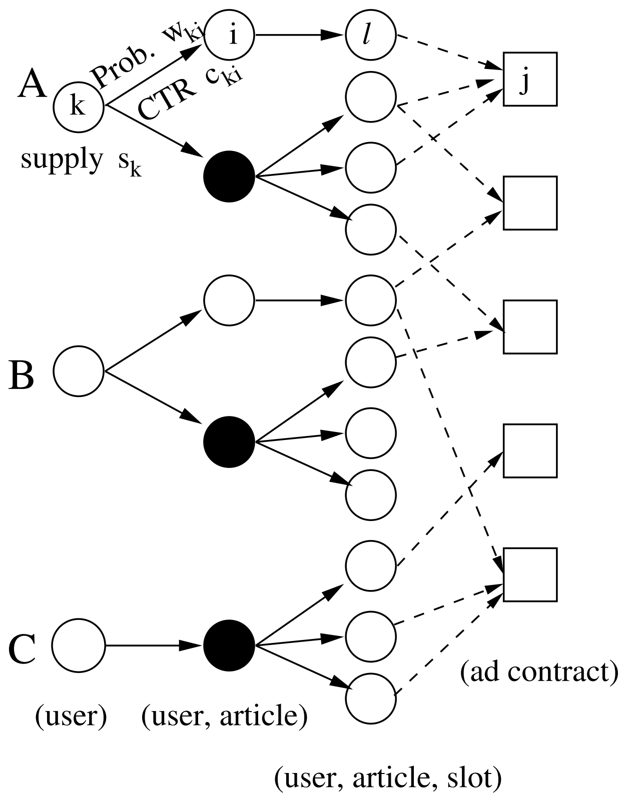


Fig. 2. Traffic shaping graph linking forecasted users and ad contracts. Users A and B arrive at the portal's homepage, while C arrives directly at an article page.

occasionally use i to refer to articles and ℓ to refer to ad slots, but the context will make the difference clear. Let $\Gamma(x)$ represent the neighbors of node x; this encodes the structure of the graph. Each user k has an associated sample size representing the expected number of users of this "type" in the forecast time period, which we call the *supply* s_k. The click-through rate for user k on a displayed article i is assumed to be known, and is denoted by c_{ki}. Each ad contract j requires a certain number of impressions to be delivered to it during the forecast period in order to satisfy its guaranteed contract; call this the *demand* d_j. An ad "underdelivers" if it receives fewer than d_j impressions. The goal is to find the optimal traffic shaping probabilities w_{ki} so that total underdelivery is minimized.

To compute underdelivery, we must understand how traffic "flows" from the user nodes (the supply) to the ad contract nodes (the demand). Consider a path $k \to i \to \ell \to j$ from user k to ad contract j. The traffic at k that flows to i is $s_k w_{ki} \mathrm{c}_{ki}$. This traffic sees every ad slot present on the article i, so the traffic at ℓ is also $s_k w_{ki} \mathrm{c}_{ki}$. Assuming the fraction of traffic at ℓ that is shown ad j is $\phi_{\ell j}$, we find that the total flow along the entire path is $s_k w_{ki} \mathrm{c}_{ki} \phi_{\ell j}$. Summing over all paths ending in any ad j gives the total flow reaching j, from which its underdelivery can be computed. Let U_i and L_i be pre-specified upper and lower bounds on w_{ki}.

Then, a basic optimization setup might be the following:

$$\begin{aligned}
&\text{Minimize} \sum_j u_j \text{ such that} \qquad (1)\\
&\sum_{\ell \in \Gamma(j)} s_k w_{ki} \mathbf{c}_{ki} \phi_{\ell j} + u_j \geq d_j \quad \forall j\\
&L_i \leq w_{ki} \leq U_i \quad \forall i\\
&\sum_{i \in \Gamma(k)} w_{ki} \leq 1 \quad \forall k\\
&\sum_{j \in \Gamma(\ell)} \phi_{\ell j} \leq 1 \quad \forall \ell\\
&u_j, w_{ki} \geq 0
\end{aligned}$$

The first constraint essentially defines the underdelivery term u_j as the difference between the demand d_j and the total delivered impressions $\sum_{\ell \in \Gamma(j)} s_k w_{ki} \mathbf{c}_{ki} \phi_{\ell j}$. Recall that there is a unique k and i for each ℓ, so the first term of the first constraint is well-defined. The second constraint allows the portal's publisher to exercise fine-grained control on the traffic shaping probabilities. For example, L_i could be set to 0.95 for the article i with the highest predicted CTR, thus ensuring that the overall CTR of the shaped traffic is close to the maximum CTR. The third constraint merely states that only available traffic can be shaped; the fourth constraint serves a similar purpose.

While this is a clean and intuitive formulation, it is nonetheless problematic. Traffic arriving at the (user, article) layer gets *multiplied* by the number of ad slots as it flows to the next layer, precluding the modeling of this as a "max-flow" problem. We also have to solve for two sets of unknowns w_{ki} and $\phi_{\ell j}$. Note that these two unknowns appear as a *product* in the first constraint; hence, the optimization is not stated as a linear program that can be solved simply. Even if we could solve for both, online reconstruction of w_{ki} in this setting appears to be difficult. The solution to this problem comes from the following insight.

LEMMA 2.1. *Suppose $\{w^*_{ki}, \phi^*_{\ell j}\}$ is an optimal solution for problem 1. Define $z_{\ell j} = \frac{w^*_{ki}\phi^*_{\ell j}}{\max_\ell \sum_{j \in \Gamma(\ell)} \phi^*_{\ell j}}$, $w'_{ki} = \max_\ell \sum_{j \in \Gamma(\ell)} z_{\ell j}$, and $\phi'_{\ell j} = \frac{z_{\ell j}}{\max_\ell \sum_{j \in \Gamma(\ell)} z_{\ell j}}$. Then, $\left\{w'_{ki}, \phi'_{\ell j}\right\}$ is also an optimal solution.*

PROOF. We may assume without loss of generality that $\max_\ell \sum_{j \in \Gamma(\ell)} \phi^*_{\ell j} > 0$; if not, some $\phi_{\ell j}$ can be increased infinitesimally (note that this can never increase any underdelivery u_j). Thus, $z_{\ell j}$ is well-defined. It may easily be determined that $\{w'_{ki}, \phi'_{\ell j}\}$ satisfy all the constraints while having the same objective as $\{w^*_{ki}, \phi^*_{\ell j}\}$. □

This lemma allows us to create an optimal solution $\{w'_{ki}, \phi'_{\ell j}\}$ where both are functions of $z_{\ell j}$. Thus, we can reformulate problem 1 as an optimization problem of just the single set of unknowns $z_{\ell j}$.

To gain an intuition about the physical meaning of $z_{\ell j}$, let us analyze the the flow of traffic through the middle layers of the graph. The total fraction of traffic at ℓ that is served some ad is $\psi_\ell = \sum_{j \in \Gamma(\ell)} \phi_{\ell j}$; note that $\psi_\ell \leq 1$, and is strictly less than 1 if there are no matching ads or the demands of the matching ads are already satisfied. Hence, the total flow through ℓ is simply $s_k w_{ki} \mathbf{c}_{ki} \sum_{j \in \Gamma(\ell)} \phi_{\ell j}$. Working backwards through the graph, we can now compute the total flow through article i. Every user who arrives at

i is exposed to all of its ad slots, and if an ad is always shown in each slot then the flow at i equals the flow at *each* of its ad slots. However, ad slots can occasionally be empty, and hence have different flows. Suppose there are two slots $\ell 1$ and $\ell 2$ available on article i, with $\psi_{\ell 2} < \psi_{\ell 1}$. Then, in the optimal solution, a user arriving at article i will see ads either on both ℓ_1 and ℓ_2, or only on ℓ_1, but never only on ℓ_2. Generalizing this to multiple ad slots, it is easily seen that the total traffic at i that sees ads is the *maximum* of the traffic through each of its ad slots, i.e., $s_k w_{ki} \mathrm{c}_{ki} \max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} \phi_{\ell j}$.

The factor $\max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} \phi_{\ell j}$ can be thought of as the "efficiency" of ad serving at article i; for every unit of traffic arriving at i, this fraction is shown some ad. Thus, $\phi_{\ell j} / \max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} \phi_{\ell j}$ is the fraction of traffic that sees ad j in slot ℓ, normalized by the fraction of traffic at article i that sees any ads at all. Multiplying this by w_{ki} gives $z_{\ell j}$; thus, $\mathrm{c}_{ki} z_{\ell j}$ is the normalized fraction of incoming users at k that see ad j.

We are now in a position to formally restate the optimization problem in terms of $z_{\ell j}$. Define $s_\ell = s_k \mathrm{c}_{ki}$; this is the supply that would be available at ℓ if all the user traffic at k had been sent its way, i.e., along the path $k \to i \to \ell$ (recall that the entire path is set when we specify ℓ). Let $h_{\ell j}(z_{\ell j})$ be a continuously-differentiable strictly convex penalty function; it is a technical device that ensures uniqueness of the solution and can be used to bias the solution (e.g., away from extremes, such as showing only one particular ad on some ad slot), but it can also be made arbitrarily small so that the underdelivery penalty dominates. Then, replacing w_{ki} with $\max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} z_{\ell j}$, we get the reformulated optimization problem:

$$\text{Minimize} \sum_j u_j + \sum_\ell \sum_{j\in\Gamma(\ell)} h_{\ell j}(z_{\ell j}) \text{ such that} \tag{2}$$

$$\sum_{\ell\in\Gamma(j)} z_{\ell j} s_\ell + u_j \geq d_j \quad \forall j$$

$$L_i \leq \max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} z_{\ell j} \leq U_i \quad \forall i$$

$$\sum_{i\in\Gamma(k)} \max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} z_{\ell j} \leq 1 \quad \forall k$$

$$u_j, z_{\ell j} \geq 0$$

We note here that problem 2 differs from problem 1 in one minor way: the total fraction of shaped traffic is allowed to be less than 1 (compare the second-last constraint in both problems). This is a simplification device that ensures that the intermediate solutions found by our algorithm are always feasible. If $\sum_i w_{ki} < 1$ at the optimal, then article i is picked for display with probability w_{ki}, and no article is displayed with probability $1 - \sum_i w_{ki}$, so the probabilistic interpretation of w_{ki} remains valid.

SUMMARY. Conceptually, the arrival of a user at the portal triggers two steps: (1) the (pre-computed) forecasted graph is augmented with nodes corresponding to this user (i.e., extra nodes in the first three layers), which are connected to available ad contracts, and (2) the optimization problem 2 on this augmented graph is solved, yielding the optimal traffic shaping probabilities for this user. The optimization is convex, and hence can be solved in time that scales linearly with the graph size. However, the graph itself can be huge; it has of the order of users $\times$ articles $\times$ slots $+$ contracts nodes. This is dominated by the first term, since a large sample of users is desirable to better represent the full distribution of forecasted traffic. Such a huge problem is impossible to solve at runtime for every incoming user. The next section presents our solution.

2.3. Online Reconstruction

Computation of traffic shaping probabilities at runtime presents a thorny problem. Time and space considerations prevent us from loading the entire graph into memory and solving the optimization, but if the probabilities are computed without reference to the full graph, all guarantees regarding underdelivery are lost. The optimal probabilities can, nonetheless, be computed by using a small cache that captures essential information from the graph, as discussed below.

The basic idea is to split up the optimization in two stages. The first stage is offline; the entire graph is loaded into memory, the optimization problem is solved, and a set of by-products (the optimization "duals") are cached. Then, at runtime, the second stage *reconstructs* the traffic shaping probabilities for the incoming user using just the cached duals and the user's characteristics. Only one dual per ad contract is required, making this a very fast and space-efficient solution. The offline solution is recomputed periodically to ensure that the cached duals remain in sync with their correct values; we find that recomputing every 15 minutes is sufficient.

We first convert the optimization problem 2 into an equivalent form that turns out to be easier for online reconstruction. We introduce an auxiliary variable v_i which, at the optimal solution, should equal $\max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} z_{\ell j}$. Then, we have the following form for the optimization:

$$\text{Minimize } \sum_j u_j + \sum_\ell \sum_{j\in\Gamma(\ell)} h_{\ell j}(z_{\ell j}) \tag{3}$$
$$\sum_{\ell\in\Gamma(j)} z_{\ell j} s_\ell + u_j \geq d_j \quad \forall j \quad \text{(dual } \alpha_j\text{)} \tag{4}$$
$$\sum_{j\in\Gamma(\ell)} z_{\ell j} \leq U_i \quad \forall \ell \qquad \text{(dual } \tau_\ell\text{)} \tag{5}$$
$$\sum_{j\in\Gamma(\ell)} z_{\ell j} \leq v_i \quad \forall \ell \qquad \text{(dual } \sigma_\ell\text{)} \tag{6}$$
$$v_i \geq L_i \quad \forall i \qquad \text{(dual } \kappa_i\text{)} \tag{7}$$
$$\sum_{i\in\Gamma(k)} v_i \leq 1 \quad \forall k \qquad \text{(dual } \mu_k\text{)} \tag{8}$$
$$u_j, z_{\ell j} \geq 0$$

The first (offline) stage, where we solve the above convex problem on the forecasted graph (Fig. 2), can be accomplished using any off-the-shelf solver, so it is not discussed any further. Once solved, we cache the duals α_j. Then, at runtime, when a new user arrives at the portal, the second stage builds the graph *only for this user* (i.e., Fig. 1) but also loads in the cached α_j values for all ad contracts. The traffic shaping probabilities must be reconstructed using only this information.

The duals are connected to the optimal solution via the Karush-Kuhn-Tucker (KKT) conditions. Define the function $g_{\ell j} = (h'_{\ell j})^{-1}$, where $h'_{\ell j}$ represents the first derivative of $h_{\ell j}$. Since $h_{\ell j}$ is continuously differentiable and strictly convex, $g_{\ell j}$ is well-defined, smooth, and monotonically increasing. In the following, we assume that it is also piecewise linear; if not, it can always be approximated arbitrarily well by a piecewise linear function. Then, at the optimal solution $z_{\ell j}$, the KKT conditions yield the following (after some algebraic manipulation):

$$z_{\ell j} = \max\{0, g_{\ell j}(\alpha_j s_\ell - \tau_\ell - \sigma_\ell)\} \tag{9}$$
$$\mu_k = \sum_{\ell\in\Gamma(i)} \sigma_\ell \text{ (for all } \{i \mid i\in\Gamma(k), v_i > L_i\}) \tag{10}$$
$$\tau_\ell \geq 0, \text{with equality if } \sum_{j\in\Gamma(\ell)} z_{\ell j} < U_i \tag{11}$$
$$\sigma_\ell \geq 0, \text{with equality if } \sum_{j\in\Gamma(\ell)} z_{\ell j} < v_i \tag{12}$$

From these equation, we prove a series of claims that motivate our online reconstruction algorithm.

CLAIM 1. *$z_{\ell j}$ is a decreasing function of* $(\sigma_\ell + \tau_\ell)$.

PROOF. The proof follows from the fact that $g_{\ell j}$ is a monotonically increasing function of its argument $(\alpha_j s_\ell - \sigma_\ell - \tau_\ell)$ and α_j and s_ℓ are known (hence fixed) during online reconstruction. □

CLAIM 2. $\sum_{j\in\Gamma(\ell)} z_{\ell j} = \min\{U_i, \sum_{j\in\Gamma(\ell)} \max\{0, g_{\ell j}(\alpha_j s_\ell - \sigma_\ell)\}\}$.

PROOF. Let $X = \sum_{j\in\Gamma(\ell)} \max\{0, g_{\ell j}(\alpha_j s_\ell - \sigma_\ell)\}$. Then, we have:

$$\sum_j z_{\ell j} < U_i \Rightarrow \tau_\ell = 0 \Rightarrow \sum_j z_{\ell j} = X, \tag{13}$$

where the two implications come from Eqs. 11 and 9 respectively. Now, suppose $X \geq U_i$. If $\sum_j z_{\ell j} < U_i$, then by Eq. 13, $\sum_j z_{\ell j} = X \geq U_i$, leading to a contradiction. Hence, $\sum_j z_{\ell j} \geq U_i$. However, $\sum_j z_{\ell j} \leq U_i$ from Eq. 5. Hence, we find that $\sum_j z_{\ell j} = U_i$. Conversely, suppose $X < U_i$. Note that $\sum_j z_{\ell j} = \sum_j \max\{0, g_{\ell j}(\alpha_j s_\ell - \sigma_\ell - \tau_\ell)\} \leq X$ from claim 1, since $\tau_\ell \geq 0$. Thus, $\sum_j z_{\ell j} \leq X < U_i$, which implies $\sum_j z_{\ell j} = X$ by Eq. 13. Thus, we have: $\sum_j z_{\ell j} = \min\{U_i, X\}$, as desired. □

Hence, $z_{\ell j}$ can essentially be treated as a decreasing function of σ_ℓ that is clipped above to ensure that $\sum_j z_{\ell j} \leq U_i$. Thus, if we define $\hat{g}_\ell(x) = \min\{U_i, \max\{0, \sum_{j\in\Gamma(\ell)} g_{\ell j}(\alpha_j s_\ell - x)\}\}$, then $\sum_j z_{\ell j} = \hat{g}_\ell(\sigma_\ell)$.

CLAIM 3. $\sum_j z_{\ell j} \leq \hat{g}_\ell(0)$.

PROOF. The function $\hat{g}_\ell$ is monotonically decreasing and piecewise linear, from the properties of $g_{\ell j}$. The claim follows. □

At optimality, $v_i = \max_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} z_{\ell j}$. Let the *max-set* $\mathcal{M}_i$ denote those ℓ that attain this maximum value v_i: $\mathcal{M}_i = \{\ell \in \Gamma(i) \mid \sum_{j\in\Gamma(\ell)} z_{\ell j} = v_i\}$. In other words, the max-set is the set of (user, article, ad slot) triplets which consume all the traffic that flows in through their parent (user, article) node; recall that some ad slots might remain empty and hence not carry all the incoming flow.

CLAIM 4. *σ_ℓ is non-zero only for $\ell \in \mathcal{M}_i$.*

PROOF. This is just a restatement of Eq. 12. □

CLAIM 5. $\hat{g}_\ell(0) \geq v_i \Leftrightarrow \ell \in \mathcal{M}_i$.

PROOF. $\ell \notin \mathcal{M}_i \Rightarrow \sigma_\ell = 0 \Rightarrow \sum_j z_{\ell j} = \hat{g}_\ell(0) \Rightarrow \hat{g}_\ell(0) < v_i$. Hence, $\hat{g}_\ell(0) \geq v_i \Rightarrow \ell \in \mathcal{M}_i$. Conversely, $\ell \in \mathcal{M}_i \Rightarrow \sum_j z_{\ell j} = v_i$, but $\sum_j z_{\ell j} \leq \hat{g}_\ell(0)$ by claim 3, so $\ell \in \mathcal{M}_i \Rightarrow \hat{g}_\ell(0) \geq v_i$. □

The online reconstruction algorithm initially sets all $\sigma_\ell = 0$, implying that all $z_{\ell j}$ are at their maximum possible values (claim 3) and underdelivery is as low as possible. If this violates constraint 8, we must reduce v_i, for which we must reduce $z_{\ell j}$ for all $\ell \in \mathcal{M}_i$ (by definition of $\mathcal{M}_i$). This is accomplished by increasing the corresponding σ_ℓ values (claim 1). However, increases in σ_ℓ must account for two factors: every ℓ that is in the current max-set must continue to be in the max-set after v_i decreases (claim 5), and the total sum $\sum_{\ell\in\Gamma(i)} \sigma_\ell$ must remain the same for all $\{i \in \Gamma(k) | v_i > L_i\}$ (Eq. 10). The following lemma shows how σ_ℓ should be changed to satisfy these conditions.

LEMMA 2.2. *For all $\ell \in \mathcal{M}_i$, the rate of change of σ_ℓ that satisfies both of the above conditions is given by:* $\sigma'_\ell \propto \frac{1/\hat{g}'_\ell}{\sum_{\ell' \in \mathcal{M}_i} 1/\hat{g}'_{\ell'}}$, *where* $\hat{g}'_\ell(x) = \lim_{h \downarrow 0}(\hat{g}_\ell(x+h) - \hat{g}_\ell(x))/h$ *is the right derivative of* $\hat{g}_\ell$.

PROOF. Consider an infinitesimal increase $\Delta\sigma_\ell$ in σ_ℓ. This decreases $\sum_j z_{\ell j}$ by $\hat{g}'_\ell \Delta\sigma_\ell$. For ℓ to remain in the max-set, v_i must decrease by the same amount, which means that every $\ell' \in \mathcal{M}_i$ must decrease by this amount: $\hat{g}'_\ell \Delta\sigma_\ell = \hat{g}'_{\ell'} \Delta\sigma_{\ell'} = \ldots = k_i$ for some k_i that is fixed for all $\ell \in \mathcal{M}_i$. Thus, $\Delta\sigma_\ell = k_i/\hat{g}'_\ell$, and the total change in σ_ℓ values connected to some (user, article) pair i is $\sum_{\ell \in \Gamma(i) \cap \mathcal{M}_i} \Delta\sigma_\ell = k_i \sum_{\ell \in \Gamma(i)} 1/\hat{g}'_\ell$. By Equation 10, the total change should be identical for all $\{i \in \Gamma(k) | v_i > L_i\}$: $k_i \sum_{\ell \in \Gamma(i) \cap \mathcal{M}_i} 1/\hat{g}'_\ell = k_{i'} \sum_{\ell' \in \Gamma(i') \cap \mathcal{M}_{i'}} 1/\hat{g}'_{\ell'} = \ldots = \phi_k$ for some ϕ_k. Thus, we can infer that $\Delta\sigma_\ell = \frac{1/\hat{g}'_\ell}{\sum_{\ell' \in \mathcal{M}_i} 1/\hat{g}'_{\ell'}} \phi_k$, proving the claim. □

Thus, the algorithm should keep increasing σ_ℓ for $\mathcal{M}_i$ at the rate specified in Lemma 2.2, until either constraint 8 is satisfied, or v_i reaches its minimum allowed value L_i, or the max-set changes. The last condition occurs when v_i is reduced to the point where it equals the ℓ with the "second-best" value of $\sum_j z_{\ell j}$; note that since this second-best ℓ was not in the max-set earlier, we have $\sum_j z_{\ell j} = \hat{g}_\ell(0)$ by claim 3. Thus, the sequence of ℓ's that join the max-set can be pre-computed. Algorithm 1 provides the details.

COMPLEXITY. At runtime, the nodes and edges corresponding to the new user have to be generated. Assuming that the pool of articles available at any time is bounded, and each article has a bounded number of ad slots, the time and space required for this graph is constant. Thus, space complexity is dominated by the cache size; assuming the number of ads that get connected to ad slots is $|A|$, this needs $O(|A|)$ space. Each iteration of the algorithm computes $z_{\ell j}$, which can take $O(|A|)$ time. The number of iterations is bounded by the number of "breakpoints" in the piecewise-linear function $\hat{g}_\ell$, plus a constant. If $\hat{g}_\ell$ has B_ℓ breakpoints, the number of iterations is $O(\sum_\ell B_\ell)$, giving a total time complexity of $O(|A| \sum_\ell B_\ell)$. This is far smaller than the $O(\text{graph-size})$ complexity of solving the entire optimization at run-time.

3. EXPERIMENTS

The previous section described our algorithm to optimally couple a portal's homepage with its ad-serving engine. In this section, we will empirically demonstrate that this coupling leads to a significant reduction in underdelivery as compared to competing approaches. Before presenting the results, we first discuss the data and the baseline algorithms.

3.1. Data

Our experiments were conducted on historical traffic logs from April, 2011 of a large web portal. The entire website can be broadly divided into several sections (e.g., Sports, News, etc.). New articles are created by editors periodically, and while their content is varied, there are always some articles from each section. Reliable estimates of CTR are available via a separate standalone module, as are forecasts of future traffic and ad contracts. Experiments were run on a random subset of nearly $25K$ user nodes and $100K$ ad contracts (these are not necessarily representative of the entire dataset).

3.2. Baseline Algorithms

Any traffic shaping algorithm must answer two questions: (a) *article selection:* how to pick articles to show to users, and (b) *ad serving:* which ads to serve on which ad slots,

ALGORITHM 1: Traffic shaping reconstruction

```
Pre-process: Compute the breakpoints $B_\ell$, where the function $\hat{g}'_\ell$ changes values
                              {$\hat{g}'_\ell$ is piecewise constant since $\hat{g}_\ell$ is piecewise linear}
Init: Set $\sigma_\ell = 0$ for all $\ell$
Set $z_{\ell j}$ = Compute-Z($\sigma_\ell$); set $z_\ell = \sum_{j \in \Gamma(\ell)} z_{\ell j}$, and $v_i = \max_{\ell \in \Gamma(i)} z_\ell$.
while $\sum_{i \in \Gamma(k)} v_i > 1$ do
  Set $\mathcal{M}_i = \{\ell \mid \ell \in \Gamma(i), z_\ell = v_i, v_i > L_i\}$
  Set $\mathcal{M} = \bigcup_i \mathcal{M}_i$
  if $\mathcal{M}$ is empty then
    Return infeasible
  end if
  Set $v_i^{(2)} = \max\{z_\ell \mid \ell \in \Gamma(i), z_\ell < v_i\}$          {"second-best" $z_\ell$ for a given $i$}
  for all $\ell \in \mathcal{M}$ do
    {$e(\ell)$ is the distance to the closest possible $\sigma$ value where (a) $\hat{g}'_\ell$ changes value, or (b)
    another $\ell'$ would have to be added to $\mathcal{M}$, or (c) $L_i$ is reached}
    $e(\ell) = \min\{\{\sigma \mid \sigma \in B_\ell, \sigma > \sigma_\ell\}, \hat{g}_\ell^{-1}(v_i^{(2)}), \hat{g}_\ell^{-1}(L_i)\} - \sigma_\ell$
  end for
  Set $\phi = \min_{\ell \in \mathcal{M}} \left\{ e(\ell) \times \frac{\sum_{\ell' \in \mathcal{M}_i} 1/\hat{g}'_{\ell'}}{1/\hat{g}'_\ell} \right\}$
  Set $\sigma_\ell = \sigma_\ell + \phi \times \frac{1/\hat{g}'_\ell}{\sum_{\ell' \in \mathcal{M}_i} 1/\hat{g}'_{\ell'}}$
  Set $z_{\ell j}$ = Compute-Z($\sigma_\ell$); set $z_\ell = \sum_{j \in \Gamma(\ell)} z_{\ell j}$, and $v_i = \max_{\ell \in \Gamma(i)} z_\ell$.
end while
Output: Traffic shaping fractions $w_{ki} = \frac{\max_{\ell \in \Gamma(i)} z_\ell}{\sum_{i' \in \Gamma(k)} \max_{\ell' \in \Gamma(i')} z'_\ell}$
```

ALGORITHM 2: Compute-Z

```
Input: $\sigma_\ell$
if $\sum_j \max\{0, g_{\ell j}(\alpha_j s_\ell - \sigma_\ell)\} > U_i$ then
  Set $\tau_\ell$ such that $\sum_j \max\{0, g_{\ell j}(\alpha_j s_\ell - \tau_\ell - \sigma_\ell)\} = U_i$
else
  Set $\tau_\ell = 0$
end if
Set $z_{\ell j} = \max\{0, g_{\ell j}(\alpha_j s_\ell - \tau_\ell - \sigma_\ell)\}$
Output: $z_{\ell j}$
```

if the user does click on the displayed article. While heuristics are easily constructed for article selection, designing alternate ad-serving solutions is quite non-trivial. Still, we constructed alternative solutions for both problems, leading to the following three baselines.

GREEDY SHAPING (GREEDY). Here, we implement a greedy alternative to both the article selection and ad serving problems. For ad serving, we find a *fraction* f_j for each ad contract j such that if j gets an f_j fraction of the supply at all the ad slots connected to it, then its demand d_j will be satisfied (unless the total connected supply is insufficient to meet the demand). These fractions f_j act as the counterparts of the duals α_j in our algorithm. To compute f_j values, we start by finding f_j for one contract, removing an f_j-fraction of the supply from each connected ad slot, and repeating this process for the next ad contract, until all contracts are processed. Ad contracts which do not have sufficient connected supply will underdeliver, and the underdelivery UD_j for each ad j is computed and stored. Note that supply forecasts for all (user, article, ad slot) triples are required as *input* to the algorithm; that is, traffic shaping is *not* being

Table I. Gains via traffic shaping

Algorithm	**Reduction in underdelivery**		**Avg. CTR**
	Number of contracts	Lift in impressions to underdelivering ads	(normalized)
CTR+Greedy	$14K$	—	—
CTR+Opt	$12K$	2.5	1.0
Greedy	$14K$	1.83	0.87
Full	$14K$	2.96	0.90
Constr	$14K$	2.57	0.96

taken into account in the forecasting or in the computation of the f_j values. Article selection at runtime is also greedy: the article selected for display is the one whose ad slots can show the most underdelivering ads. In other words, we select article i with the highest $c_{ki} \sum_{\ell\in\Gamma(i)} \sum_{j\in\Gamma(\ell)} f_j \mathrm{UD}_j$.

CTR-ONLY WITH GREEDY (CTR+GREEDY). This baseline decouples article selection from ad serving. The portal always selects the article with the highest expected CTR, and no effort is made to increase downstream ad revenue. Ad-serving is done using the same greedy method outlined above.

CTR-ONLY WITH UNDERDELIVERY OPTIMIZATION (CTR+OPT). Here, we pair the article selection of CTR+Greedy with our optimization-based ad-serving. More precisely, the article with the highest expected CTR is always selected. The ad-serving mechanism remains the same as in our algorithm, and uses the same underdelivery-minimization objective. In essence, the ad-serving system acts as if the user had directly arrived at the article page (recall the example of user C in Figure 2 earlier).

These baselines are compared against our traffic shaping algorithm via a "replay" experiment, where historical traffic patterns were presented as input to the algorithm, which had to decide both the article summaries and the ads to show to the user. Assuming correct CTR estimates, the number of impressions delivered to the various ad contracts is tracked. At the end of the replay, the number of underdelivering contracts and the total shortfall in impressions are computed and compared across algorithms.

3.3. Usefulness of Traffic Shaping

We experimented with two versions of our algorithm:

— *Full shaping (Full):* No constraints are placed on the shaping probabilities, i.e., we used trivial lower and upper bounds $L_i = 0$ and $U_i = 1$.

— *Constrained shaping (Constr):* Here, the traffic shaping algorithm is constrained to show every user the article with the highest CTR with at least 90% probability, leaving only 10% of the traffic available for shaping.

For each case, we solve the offline optimization to get the α_j duals that must be used in online reconstruction. For the baselines, we also solve for the fractions f_j corresponding to the ad contracts. This also yields, as a by-product, the number of underdelivering ad contracts (around $14K$ out of $100K$ contracts).

For each algorithm, we report three numbers. The first is the number of underdelivering ads that are delivered impressions thanks to the traffic shaping algorithm. The second measures the total impressions delivered to underdelivering ads. For reasons of confidentiality, we present this not in terms of the actual number of such impressions but rather as a *lift* in the impressions by the algorithm over the impressions delivered by CTR+Greedy. The third number we present is the average CTR of the algorithm, again normalized by that of CTR+Greedy. Table I shows the performance of all the algorithms.

We can make the following observations:

(1) Both Full and Constr deliver significantly more impressions to underdelivering ads as compared to the baselines.
(2) Overall, all algorithms are able to deliver impressions to the bulk of the underdelivering ad contracts.
(3) The average CTR is highest for CTR+Greedy and CTR+Opt, as expected; in both, article selection is done solely to maximize CTR. However, Full is able to achieve 90% of this CTR even though it does not enforce any constraints on the traffic shaping probabilities. When we do constrain the algorithm to select the maximum-CTR article for at least 90% of the traffic, the average CTR is actually within 96% of the maximum (note that the maximum-CTR article can depend on user characteristics, and can be different for different user nodes). This implies that the "shaped" traffic was shown articles that had a CTR of about 60% of the maximum. It is in fact likely that articles selected by our shaping algorithm will have moderately high CTR: if CTR is too low, the number of users clicking on the article, and hence the total traffic that can be delivered to underperforming ads, will be too low to be worthwhile.

3.4. Effect of constraints

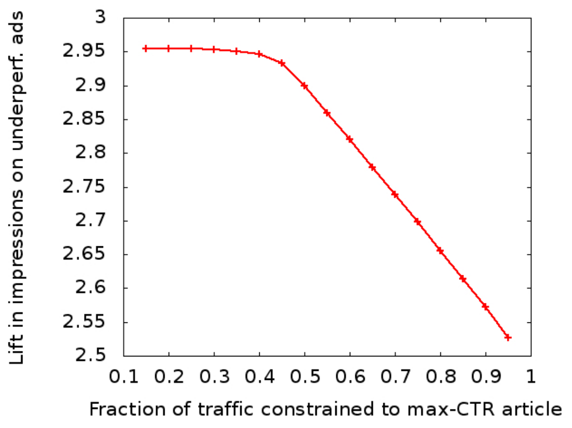
To better understand the trade-offs involved in constraining traffic shaping to the maximum-CTR articles, we analyzed a range of constraint values (i.e., the L_i values from our optimization). Figure 3(a) and (b) plot the lift in delivered impressions and the average CTR as functions of the constraint. As expected, when the algorithm is forced to display the maximum-CTR article to a higher fraction of traffic, the lift in impressions delivered to underperforming ad contracts is reduced simply because less traffic is available for shaping. Correspondingly, the average CTR increases. A careful analysis also reveals the existence of three distinct regimes:

(1) $L_i \leq 0.25$: Both plots are flat when a 0.25 or smaller fraction of the traffic is constrained, implying that the optimal unconstrained solution (Full) sends about 25% of the traffic to the maximum-CTR article anyway.
(2) $0.25 < L_i \leq 0.4$: Here, we see that the average CTR starts climbing, implying that the Full was shaping this traffic to articles with lower CTR. However, the change in lift in this range is marginal. Thus, even though the articles being picked by Full were connected to more underdelivering contracts, their lower CTR did not allow this advantage to be translated into significant lift. Clearly, the portal's publisher should send at least 40% of incoming traffic to the best article; this maximizes user engagement (as measured by CTR) while retaining practically all the advantages of the unconstrained traffic shaping solution.
(3) $L_i > 0.4$: When more than 0.4 fraction of the traffic is constrained, the effect is linear over a broad range of constraint values. Every unit of traffic above the 40% bar that is sent to the maximum-CTR article is a unit of traffic that is shown significantly fewer underdelivering ads.

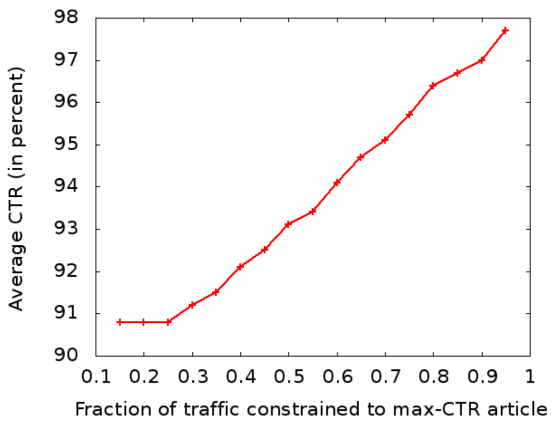
Finally, we note two points. First, impressions were delivered to all $14,000$ underperforming contracts for all values of L_i shown in the plots. Thus, our traffic shaping algorithm remains "fair" to all contracts requiring extra impressions. Second, even with unconstrained traffic shaping, the average CTR decreases only to 90% of the maximum possible, while yielding an almost threefold lift in reducing underdelivery. This is an emphatic justification for the usefulness and power of traffic shaping.

4. GENERAL FORM

Our proposed algorithm was designed specifically to address the objective function and constraints that matter for the traffic shaping problem, but its basic underlying ideas



(a) Lift in impressions



(b) Average CTR

Fig. 3. Variation in (a) lift in impressions and (b) average CTR, as the traffic shaping algorithm is constrained to send different fractions of incoming traffic to the maximum-CTR article.

might be applicable in many other settings. We devote this section to a presentation of this general methodology.

Consider the following problem:

$$\text{(ORIG)} \qquad \text{Minimize} \qquad H(x) \tag{14}$$

$$\text{s.t. } \forall j \in \mathcal{J} \;\; c_j \cdot x + d_j \leq 0, \tag{15}$$

where the objective $H(x)$ is strictly convex and separable (i.e. $H(x) = \sum_i H_i(x_i)$), $\mathcal{J}$ is a set of indices for our constraints, and c_j, d_j, and x are all vectors. We will occasionally abuse notation, referring to $\mathcal{J}$ as the set of constraints, rather than their indices.

In the *offline phase*, we solve the above problem, ensuring that we find an optimal set of duals. That is, we find x^* and duals α_j satisfying the KKT-conditions:

$$\nabla H(x^*) + \sum_{j \in \mathcal{J}} \alpha_j c_j = 0 \qquad \textit{Stationarity} \tag{16}$$

$$\forall j \in \mathcal{J}, \;\; \alpha_j(c_j \cdot x^* + d_j) = 0 \quad \textit{Complementary slackness} \tag{17}$$

$$\forall j \in \mathcal{J}, \;\; c_j \cdot x^* + d_j \leq 0 \qquad \textit{Primal feasibility} \tag{18}$$

$$\forall j \in \mathcal{J}, \;\; \alpha_j \geq 0 \qquad \textit{Dual feasibility} \tag{19}$$

For the types of problems we are interested in, it is impractical or impossible to store the full primal solution. Instead, we wish to store a small amount of information (in the form of a subset of the dual values) and reconstruct the primal solution online, as needed.

Our method works on problems that are *shatterable* into small subproblems, a concept that we now define: Denote the set of indices of x by $\mathcal{I}$; thus, we may think of the variables in our problem as x_i for $i \in \mathcal{I}$. Consider the graph induced by the set of constraints in our problem: Each node corresponds to a variable x_i for $i \in \mathcal{I}$, and an edge (i, i') exists if x_i and $x_{i'}$ appear in a constraint together. (Here, we only consider x_i to appear in constraint $c_j \cdot x + d_j \leq 0$ if $c_{ij} \neq 0$, where c_{ij} is the i-th entry in c_j.) Notice that this induced graph has a number of connected components (possibly just one), and that the set of connected components corresponds to a partition of $\mathcal{I}$.

We say a set of constraints $\mathcal{J}_{shatter} \subseteq \mathcal{J}$ *shatters* the problem into partition $\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k$ of $\mathcal{I}$ if removing the constraints $\mathcal{J}_{shatter}$ from the problem produces a graph whose connected components correspond to $\mathcal{I}_1, \ldots, \mathcal{I}_k$.

Why is this concept so useful? Consider the following problem

$$\text{Minimize} \quad G(x) \tag{20}$$
$$\text{s.t. } \forall j \in \mathcal{J} \setminus \mathcal{J}_{shatter} \;\; c_j \cdot x + d_j \leq 0, \tag{21}$$

In the case that $G(x)$ is separable (i.e., $G(x) = \sum_{i \in \mathcal{I}} G_i(x_i)$), this problem actually breaks into k smaller problems, each of the form

$$\text{Minimize} \;\; \sum_{i \in \mathcal{I}_\ell} G_i(x_i) \tag{22}$$
$$\text{s.t. } \forall j \in \mathcal{J}_\ell, \;\; c_j \cdot x + d_j \leq 0, \tag{23}$$

where $\mathcal{J}_\ell$ is the set of constraints from $\mathcal{J} \setminus \mathcal{J}_{shatter}$ for which some x_i appears for $i \in \mathcal{I}_\ell$ (i.e. the connected component corresponding to $\mathcal{I}_\ell$).

In the problem we studied in this paper, the demand constraints shattered our problem into a number of subproblems, one for each user. Thus, in the online reconstruction phase, we solved a small subproblem that depended only on the connected component involving an individual user.

Again, our technique is suited towards problems that can be shattered into appropriate subproblems. In this case, let $\mathcal{J}_{shatter}$ be the shattering constraints. After the offline phase, we store the dual values α_j for these constraints $j \in \mathcal{J}_{shatter}$ (and forget the primal solution and all other dual values).

During *online reconstruction*, if we need to find the value of some x_i with $i \in \mathcal{I}_\ell$, we simply solve a smaller problem:

$$\text{Minimize} \;\; \sum_{i \in \mathcal{I}_\ell} H_i(x_i) + \sum_{j \in \mathcal{J}_{shatter}} \alpha_j \sum_{i \in \mathcal{I}_\ell} c_{ij} x_i \tag{24}$$
$$\text{s.t. } \forall j \in \mathcal{J}_\ell, \qquad c_j \cdot x + d_j \leq 0, \tag{25}$$

where the α_j are now fixed constants that were determined in the offline phase.

In general, this problem may be solved using the same technique as the original problem, albeit much more quickly due to its smaller size. Of course, in many cases (such as the problem we study), this subproblem may be solved even more efficiently.

Finally, we need to argue that the solution produced by online reconstruction is identical to the solution of the original problem. Note that although we have solved a number of smaller problems, the solution to each of these problems together is the same as the following larger problem:

$$\text{(SUB)} \qquad \text{Minimize} \;\; H(x) + \sum_{j \in \mathcal{J}_{shatter}} \alpha_j (c_j \cdot x + d_j) \tag{26}$$
$$\text{s.t. } \forall j \in \mathcal{J} \setminus \mathcal{J}_{shatter}, \qquad c_j \cdot x + d_j \leq 0 \tag{27}$$

Notice that we have somewhat suggestively added $\sum_j \alpha_j d_j$ to the objective function. This is just a constant, so it does not affect the optimal solution. We have the following.

CLAIM 6. *The primal solution to (SUB) is identical to the primal solution of (ORIG).*

PROOF. We will prove the case for $\mathcal{J}_{shatter}$ of size 1, say $\mathcal{J}_{shatter} = \{1\}$. The proof for larger $\mathcal{J}_{shatter}$ follows by induction.

Let y^* be the optimal solution to

$$\text{Minimize} \;\; H(x) + \alpha_1(c_1 \cdot x + d_1) \tag{28}$$
$$\text{s.t. } \forall j \in \mathcal{J} \setminus \{1\} \qquad c_j \cdot x + d_j \leq 0 \tag{29}$$

Note that x^*, the optimal solution to (ORIG), is also a solution to the above problem. We wish to show that $y^* = x^*$. Suppose this is not the case, and $y^* \neq x^*$. We will show that $H(x^*) + \alpha_1(c_1 \cdot x^* + d_1) < H(y^*) + \alpha_1(c_1 \cdot y^* + d_1)$, hence y^* cannot be optimal.

Define $p(t) = H(x(t)) + \alpha_1(c_1 \cdot x(t) + d_1)$, where $x(t) = ty^* + (1-t)x^*$. (Intuitively, this is the restriction of the new objective function to the line segment from x^* to y^*.) Note that $p(0) = H(x^*) + \alpha_1(c_1 \cdot x^* + d_1)$, while $p(1) = H(y^*) + \alpha_1(c_1 \cdot y^* + d_1)$. Further, since H is strictly convex, it follows by definition that p is strictly convex as well.

Consider the derivative of p,

$$p'(t) \ = \nabla H(x(t)) \cdot (y^* - x^*) + \alpha_1 c_1 \cdot (y^* - x^*) \tag{30}$$

Hence,

$$\begin{aligned} p'(0) &= \nabla H(x^*) \cdot (y^* - x^*) + \alpha_1 c_1 \cdot (y^* - x^*) && (31)\\ &= -\textstyle\sum_{j \in \mathcal{J} \setminus \{1\}} \alpha_j c_j \cdot (y^* - x^*), && (32) \end{aligned}$$

where the second equality follows from the stationarity condition for (ORIG). Further, by the complementary slackness conditions for (ORIG), we see that for all $j \in \mathcal{J} \setminus \{1\}$, either $\alpha_j = 0$ or $c_j \cdot x^* + d_j = 0$. Since $c_j \cdot y^* + d_j \leq 0$ for $j \in \mathcal{J} \setminus \{1\}$, this implies that $-\alpha_j c_j \cdot (y^* - x^*) \geq 0$. Hence, $p'(0) \geq 0$.

Continuing, since p is strictly convex on $[0, 1]$, its derivative is strictly increasing. Hence, $p'(t) > 0$ on $(0, 1]$. That is, p is strictly increasing. Hence, $p(0) < p(1)$. So we have

$$H(x^*) + \alpha_1(c_1 \cdot x^* + d_1) < H(y^*) + \alpha_1(c_1 \cdot y^* + d_1)$$

as claimed. That is, x^* is the unique optimal solution to our modified problem. The proof follows. □

5. BACKGROUND AND RELATED WORK

To the best of our knowledge, this work is the first paper focused on delivering web content in a way that enhances short-term engagement with the user, while simultaneously enhancing the delivery guarantees of a pre-existing large-scale system.

There have been previous efforts that dealt with selecting content to increase engagement. Most of this work has been concerned with single metrics, like clickthrough rate ([Agarwal et al. 2008; Das et al. 2007]). More generally, [Yang et al. 2010] consider balancing multiple objectives, including both increasing clickthrough rate as well as those objectives associated with guaranteed delivery contracts. Recently, the work of [Agarwal et al. 2011] has taken engagement to involve balancing a variety of metrics, including potential downstream revenue. This downstream impact includes considerations like increasing the total time spent on web pages. (Users who are shown good content are likely to stay around longer.) However, these considerations were treated as static values. Even if these values are updated frequently, they fail to capture the interplay between content and ad delivery.

In contrast, our work allows the associated guaranteed ad delivery system to book ad contracts more aggressively, knowing that shortfalls in inventory can be corrected dynamically. Further, even when no over-booking happens, we can balance the allocation and delivery goals of publishers with the short-term engagement of users.

A second issue we address is how to make online decisions to deliver content in a near-optimal way. One of the main difficulties is that real systems must be robust — delivering content in a good fashion regardless of what users appear — while maintaining just a small amount of state. There has been work on content optimization that modeled this as a multi-armed bandit problem [Dudík et al. 2011; Agarwal et al. 2009]. However, this work fails to capture the interactions of content choices with a larger guaranteed ad delivery system.

Our technique utilizes a pre-existing forecast to compute an optimal solution. (Note that such a forecast must already exist in order for a guaranteed delivery system to ensure that the guarantees are deliverable.) Although this seems straightforward at first blush, it is actually only the starting point for content delivery. Storing the entire solution on each server is entirely impractical. Further, if the forecast fails to include even a single user, the server would not know what to do. To address these concerns, earlier work proposed storing the duals for a subset of the constraints [Devenur and Hayes 2009; Vee et al. 2010][5] At serving time, these duals then allow the server to reconstruct the primal solution. However, these earlier papers considered much simpler optimization problems. In the case of [Devenur and Hayes 2009], it is a linear program in which the main constraints are all upper bounds. Although [Vee et al. 2010] studies a problem with a non-linear objective, it focuses on the case of a bipartite graph. Here, the input problem is a four-layer graph, and simply storing the duals for the constraints would not have produced a usable solution. Indeed, much of the difficult technical work in this paper was re-formulating the problem in such a way that the dual formulation could compactly represent the primal in a way that was reconstructable at serving time.

6. CONCLUSIONS

Article selection and ad serving are usually solved by standalone systems, and this has meant that underdelivering ads could not be “helped”. We propose an optimization-based formulation of traffic shaping that minimizes underdelivery while ensuring that the maximum-CTR articles are still selected at least some pre-specified fraction of the time. Thus, the objective jointly optimizes for underdelivery (ad serving) and CTR (user engagement through article selection). We further give an online reconstruction algorithm that can infer the optimal traffic shaping probabilities for each incoming user using only a small cache of duals; thus, the entire forecast graph need not be loaded into memory. The degree of traffic shaping can be controlled by the publisher to conform as closely as desired to any standalone article selection solution. Empirical results on a large dataset from a real-world web portal demonstrate that a threefold reduction in underdelivery is possible with only a 10% CTR drop, or a 2.6-fold improvement with only a 4% CTR drop, clearly demonstrating the effectiveness of traffic shaping.

REFERENCES

Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. 2009. Explore/Exploit Schemes for Web Content Optimization. In *ICDM*. 1–10.

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, Nitin Motgi, Seung-Taek Park, Raghu Ramakrishnan, Scott Roy, and Joe Zachariah. 2008. Online Models for Content Optimization. In *NIPS*. 17–24.

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. 2011. Click shaping to optimize multiple objectives. In *KDD*. 132–140.

Abhinandan Das, Mayur Datar, Ashutosh Garg, and ShyamSundar Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *WWW*. 271–280.

Nikhil R. Devenur and Thomas P. Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *ACM Conference on Electronic Commerce*. 71–78.

Miroslav Dudík, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. 2011. Efficient Optimal Learning for Contextual Bandits. In *UAI*. 169–178.

Erik Vee, Sergei Vassilvitskii, and Jayavel Shanmugasundaram. 2010. Optimal online assignment with forecasts. In *EC*. 109–118.

[5]Note that [Devenur and Hayes 2009] assumes a random arrival model and computes the duals on the fly; however, we can remove the random arrival assumption if we view the duals as being computed from a forecast.

Jian Yang, Erik Vee, Sergei Vassilvitskii, John Tomlin, Jayavel Shanmugasundaram, Tasos Anastasakos, and Oliver Kennedy. 2010. Inventory Allocation for Online Graphical Display Advertising. *CoRR* abs/1008.3551 (2010).

Zenith Optimedia 2011. Quadrennial events to help ad market grow in 2012 despite economic troubles. http://www.zenithoptimedia.com/about-us/press-releases/zenithoptimedia-adspend-forecast-update-dec-2011/. (2011).