A Unified Framework for the Scheduling of Guaranteed Targeted Display Advertising under Reach and Frequency Requirements

Ali Hojjat, John Turner
Paul Merage School of Business, UC Irvine
{hojjats, john.turner}@uci.edu

Suleyman Cetintas, Jian Yang
Yahoo Labs, Sunnyvale, CA
{cetintas, jianyang}@yahoo-inc.com

Abstract

Motivated by recent trends in online advertising and advancements made by online publishers, we consider a new form of contract which allows advertisers to specify the number of unique individuals that should see their ad (*reach*), and the minimum number of times each individual should be exposed (*frequency*). We develop an optimization framework that aims for minimal under-delivery and proper spread of each campaign over its targeted demographics. As well, we introduce a *pattern*-based delivery mechanism which allows us to integrate a variety of interesting features into a website's ad allocation optimization problem which have not been possible before. For example, our approach allows publishers to implement any desired pacing of ads over time at the user level or control the number of competing brands seen by each individual. We develop a two-phase algorithm that employs column generation in a hierarchical scheme with three parallelizable components. Numerical tests with real industry data show that our algorithm produces high-quality solutions and has promising run-time and scalability. Several extensions of the model are presented, e.g., to account for multiple ad positions on the webpage, or randomness in the website visitors' arrival process.

Keywords: Online Advertising, Guaranteed Targeted Display Advertising, Reach, Frequency, Uniform Delivery, Column Generation, Cutting Stock, Quadratic Programming.

1 Introduction

Since its advent, internet advertising has drawn a lot of attention due to its interactivity, ease of customization, world-wide reach, and effective targeting abilities. This segment has grown from $9.6 billion in 2004 to $49.5 billion in 2014, exceeding all other forms of advertising such as broadcast and cable television, radio, newspaper, and consumer magazines (IAB 2015). Efficient serving of advertising is a key problem for online publishers such as Yahoo, Facebook, and Google. A large publisher may have hundreds of millions of page visits per day, and tens of thousands of concurrent advertising campaigns to manage, many of which have been booked and guaranteed well in advance. Each page visit poses a split-second opportunity to the publisher to choose one or more ads to show to the user. Even a few percent improvement in drawing the correct ad for each user can improve annual publisher revenues by tens of millions of dollars[1] while enhancing user experience.

In all existing forms of online advertising contracts, campaigns specify an aggregate impression goal or a budget limit and do not differentiate between 2 impressions of the same ad served to a

single user, or 1 impression served to each of 2 distinct users. However, industry trends show that advertisers are becoming more concerned about who they reach (Warc 2015) and traditional media measurement metrics of reach (how many unique individuals were exposed to the ad), frequency (how many times, on average, each individual was exposed to the ad), and Gross Rating Points (GRP) are increasingly being adopted by online advertisers (eMarketer 2009). Alongside the tremendous growth of online video streaming sites (such as YouTube, Netflix, etc.), video ads have gained much attention and are used to complement TV ad campaigns, which makes classic reach and frequency metrics important in designing and measuring online campaigns (eMarketer 2014). *People-based marketing* has been a popular catchphrase in the industry over the past year and advertising companies are exerting major efforts to measure and track individuals (c.f., Kattula et al. 2015). The exponential growth in the use of portable devices has made mobile advertising the fastest growing segment of online media (with 110% CAGR), and more advanced identifier technologies (such as Apple's IDFA and Google's Advertising ID) have made it easier for publishers to track individuals over time across multiple devices. Online ads are becoming more relevant and personalized than ever before, and promotion is shifting toward *storytelling* where the advertising message is broken into small bite-sized pieces. The recent case study of Adaptly (2014) on Facebook shows that creative sequencing of ads at a personal level substantially increases view-through and subscription rates.

Motivated by these industry trends, in our paper, we consider an entirely new form of advertising campaign, under what we call a *Reach and Frequency (R&F)* contract, which allows campaigns to explicitly specify the viewer demographics eligible to see their ad (*targeting*), the number of unique individuals that should see the ad (*reach*), and a required number of times that each individual should be exposed to the ad (*frequency*) for him/her to be considered as reached. The publisher receives revenue for the number of unique individuals reached at the specified frequency.

We develop an optimization model for a publisher to optimally plan and serve R&F contracts which maximizes retained revenue (i.e., minimizes under-delivery), and has several important features for both the advertiser and the publisher. First, our model produces plans that are well-dispersed within each campaign's targeted demographic (advertisers expect the publisher to not deliver the campaign to only a small, potentially easy-to-serve, subgroup of targeted users). Second, our modeling approach explicitly takes into account the user-level sequence of ads over time. This allows advertisers to implement sequenced (storyboarded) ad campaigns, as well as to specify their desired rate of re-exposure (i.e., whether impressions of an ad should be served to a user upon consecutive visits to promote recall, or evenly paced over time). Third, our model can maximize the diversity of campaigns seen by each user, or restrict the number of competing brands shown to each user (e.g., Pepsi and Coke). To the best of our knowledge, none of these user-level features are explicitly considered in the existing models for planning online advertising.

Our optimization model includes several features which make it attractive for implementation for publishers. First, it exhibits promising run-time and scales well to industry-size problems, due to the fact that each component of our model is parallelizable. Second, because of the combinatorial

explosion of targeting dimensions and the long-tailed nature of user behavior, it is prohibitive for any publisher to produce an ad delivery plan that includes every possible user type. Using duality theory, we show that a near-optimal allocation rule can be determined for user types which have never been seen before or not explicitly considered when the plan was produced.

Our paper contributes to the literature of operations research and online advertising in a variety of aspects. To the best of our knowledge, our work is the first to introduce R&F contracts and consider the optimal scheduling of online advertising under explicit reach and frequency specifications. As well, our model is the first that explicitly incorporates user-level quality metrics, such as diversity and pacing of ads over time for each user, into the publisher's ad planning problem. We introduce a new mechanism for ad serving, which we call *pattern*-based ad delivery, that pre-generates an explicit sequence of ads for each user to see over time. This mechanism is essential to our ability to plan at the user level while keeping the dimensionality of the optimization problem manageable. Our novel pattern-based method, called *Hierarchical Column Generation* (henceforth Pattern-HCG), gives rise to a fresh application of column generation in the form of an iterative algorithm with two phases and three inter-related components. We conduct a comprehensive set of tests to evaluate the performance of our methodology on real industry data obtained from *Yahoo*. Since prior work in planning online advertising is impression-based, we propose two heuristics which serve as benchmarks for our Pattern-HCG algorithm. First, we describe an adaptation of *frequency capping*, which is an existing industry practice within the context of impression-based ad planning that limits the number of times each individual is exposed to the same ad. Next, we develop a pattern-based greedy heuristic (henceforth Pattern-G) which avoids some of the computational complexities of Pattern-HCG such as the need for column generation or additional iterations for parameter-tuning. Our experiments demonstrate that Pattern-HCG achieves a 10% reduction in under-delivery compared to Pattern-G, and a 45% reduction in under-delivery compared to frequency capping.

This paper is organized as follows. We begin with an overview of the relevant literature in §2. In §3, we further elaborate on reach and frequency planning and appropriate quality metrics. To contrast our work with current practice, we describe an existing model for the planning of impression-based campaigns with several important features, as well as the frequency capping heuristic. In §4, we formally introduce how patterns can be used to serve advertising and describe our Pattern-G heuristic. In §5, we present our Pattern-HCG method. As well, we highlight structural similarities and differences between our R&F ad planning problem and the classic cutting stock problem, and point out the shortcomings of using a direct application of column generation without hierarchical decomposition. Finally, we conduct a thorough set of numerical experiments in §6 to demonstrate the performance and robustness of our methodology. Concluding remarks, insights and directions for future research appear in §7. Proofs of all theorems along with several extensions of the model and supplementary discussions are included in the appendices.

2 Literature Review

Reach and frequency are well-established marketing metrics for planning and evaluating the effectiveness of advertising campaigns. There is an extensive body of empirical research that examines the impact of ad repetition on user recall. These studies commonly agree that initial exposures to a message first increase attitude toward the product due to positive habituation (*wear-in* effect), but too many exposures lead to tedium/boredom and lower attention, and therefore decrease attitude toward the product (*wear-out* effect). The two effects produce an S-shaped response function, i.e., an inverted-U relationship between the n'th exposure and incremental message impact (see Campbell and Keller 2003 and references therein). Chandler-Pepelnjak and Song (2003) demonstrate how historical campaign performance can be used to determine the most efficient or most profitable campaign-specific frequency rates. There is also a rich literature that employs dynamic optimal control to determine the optimal rate of advertising expenditures over time in order to maximize a single advertiser's net present profit, in a finite or infinite horizon setting (see Sethi 1977, and Feichtinger et al. 1994 for comprehensive reviews). Our model does not recommend appropriate reach and frequency levels for advertisers. Instead, we take these parameters as given and solve the publisher's allocation problem which simultaneously seeks to meet all advertisers' reach and frequency requirements using the available supply of impressions.

Mathematical modeling of the ad allocation problem as a *transportation problem*, i.e., bipartite graph with supply and demand nodes that represent viewer types and ad campaigns, has been a very useful modeling approach and quite successful in practice. Langheinrich et al. (1999) is among the first to use a linear transportation problem to maximize the total click-through rate. Tomlin (2000) suggests using a nonlinear entropy term in the objective to obtain more dispersed and thus robust solutions. Chickering and Heckerman (2003) use hierarchical linear programming (LP) to produce a uniformly-spread schedule with maximum overall click-though and demonstrate the effectiveness of this approach through experiments on *msn.com*. Nakamura and Abe (2005) propose a number of improvements to the base LP formulation, including lower bounds for decision variables, importance weights for contracts, using the Gittins index in place of click-through estimates coupled with an interior-point algorithm to address the exploration-exploitation tradeoff, and clustering viewer types with similar click-through rates to increase prediction accuracy and reduce LP dimensionality. More recently, Turner (2012) uses a quadratic objective to spread impressions across viewer types, which directly minimizes the variance of the number of impressions served. Bharadwaj et al. (2012) consider CPM contracts (for which click-through does not play a role) and minimize a weighted objective composed of linear under-delivery and quadratic spreading metrics. They develop an efficient algorithm, called SHALE, to solve their formulation with minimal memory usage and better run-time than commercial solvers on industry-size instances.

Column generation (CG) is a classical method for solving mathematical programs with an exponential number of variables in which the number of positive variables in the solution is expected to be relatively small. This method has been used extensively for efficiently solving the cutting stock problem (see Gilmore and Gomory 1961), as well as problems in vehicle routing, crew/job/machine

scheduling, multi-commodity flow problems, traffic assignment, graph coloring, clustering, and many others (see Lübbecke and Desrosiers 2005, and Desaulniers et al. 2005 for thorough reviews). There are a few papers that employ CG in the context of online advertising. Abrams et al. (2008) develop a column-based formulation for the allocation of sponsored search. In their model, a column corresponds to an ordered arrangement of ads into webpage slots which will be shown to a user *all at once* when the page is loaded. The expected revenue of showing any particular arrangement is pre-calculated using generalized second price auction rules. The optimization problem determines the number of times each arrangement should be displayed in response to each search query to maximize publisher's revenue, subject to expected query inventory and the advertisers' budget. Salomatin et al. (2012) combine the planning of guaranteed and non-guaranteed advertising by allowing the arrangement (column) to contain both auction-type and guaranteed ads. They maximize total revenue collected across both types of campaigns minus any under-delivery penalties. Contrary to the above modeling approaches, columns of our model represent the sequence of ads for each user over *time*, allowing us to focus on reach and frequency as measured for each individual user over a given horizon.

Finally, a number of authors consider the **revenue optimization** of online advertising in a variety of settings (e.g., see Roels and Fridgeirsdottir 2009; Mookerjee et al. 2012; Najafi Asadolahi and Fridgeirsdottir 2014; Balseiro et al. 2014). Although every publisher's goal is revenue maximization, our focus here is on the allocative efficiency of guaranteed campaigns which, when done well, leads to high profits.

3 The Ad Allocation Problem

The general problem setting in which our problem is couched is one of matching supply with demand, where demands are known and units of supply arrive incrementally over a fixed time horizon. Formally, the publisher observes a sequence of impression arrivals $a = 1..A$ (we assume one ad-serving opportunity per arrival, which is consistent with how dynamically-generated webpages often request ads, as well as how video ads are requested from an ad server one at a time; for completeness, we also consider the case of multiple impressions per arrival in Appendix C). Each impression arrival a corresponds to a user j_a and a timestamp t_a indicating the time of the arrival. Over a fixed planning horizon (e.g., one week), the number of impression arrivals A is uncertain, as is the entire sequence $(t_1, j_1), (t_2, j_2), \ldots, (t_A, j_A)$. On the demand side, the publisher has a given set of ad campaigns, denoted $\mathcal{K}$. Each campaign $k \in \mathcal{K}$ specifies a desired reach of r_k unique users, where each user is required to see the ad f_k times (i.e., the ad's frequency) to count as being reached. We must choose which ad k to assign to each impression arrival a, bearing in mind that only a subset of users $\tilde{\Gamma}(k)$ may be matched with ad k; this is known as targeted advertising. How we assign impression arrivals to ad campaigns determines which users are counted as reached. Let $y_{ak} = 1$ when we assign ad k to arrival a, and $y_{ak} = 0$ otherwise. Then, $z_{jk} = \mathbb{1}(\sum_{\substack{a=1..A: \\ j_a=j}} y_{ak} \geq f_k)$ indicates whether or not user j was reached, and evaluates to 1 iff user j is exposed to campaign

k at least f_k times.

There are a number of objectives which are relevant in this general setting. We distinguish between *aggregate quality objectives* $Q_A(\boldsymbol{z})$ which measure the quality of the assignment as a function of *who was reached* (i.e., the z_{jk} variables), and *disaggregate quality objectives* $Q_D(\boldsymbol{y})$ which measure the quality of the assignment as a function of the *specific sequence of ads assigned to each user* (i.e., the y_{ak} variables, as the z_{jk} variables may not retain enough information to compute the disaggregate quality metric). Publishers seek both aggregate and disaggregate quality; however, aggregate quality is more important since it is closely tied to contractual obligations with direct revenue consequences. We propose a bi-criteria optimization problem with $Q_A(\boldsymbol{z})$ as the primary objective and $Q_D(\boldsymbol{y})$ as the secondary objective. Formally speaking, if we denote $Y = \{\boldsymbol{y} \in \{0,1\}^A : \sum_{k \in \mathcal{K}} y_{ak} \leq 1,\ \forall a = 1..A;\ y_{ak} = 0,\ \forall a = 1..A, k \in \mathcal{K} :\ \ j_a \notin \tilde{\Gamma}(k)\}$ as the set of feasible assignments of ads to impressions, and $Q_A^* = \max_{\boldsymbol{y} \in Y} Q_A(\boldsymbol{z}(\boldsymbol{y}))$ as the maximum value achievable for the aggregate quality objective, then we are interested in solving $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y} \in Y}\{Q_D(\boldsymbol{y}) : Q_A(\boldsymbol{z}(\boldsymbol{y})) = Q_A^*\}$. As defined, the primary (aggregate) objective dominates the secondary (disaggregate) objective; thus, no improvement in the secondary objective can be made that sacrifices the value of the primary objective.

In some cases, as we will soon see, it is important to use a randomized policy that produces different assignments y_{ak} when run multiple times on the same impression arrival sequence. In this case, the impression assignment y_{ak} as well as the reach indicator variables z_{jk} are random variables, as they are both functions of the random draws $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_A\}$ made by the policy. Formally, let $\tilde{\pi}$ denote a randomized policy that chooses $\boldsymbol{y}$ given a sequence of random draws $\boldsymbol{\xi}$, i.e., $\boldsymbol{y} = \tilde{\pi}(\boldsymbol{\xi})$, and define $\Pi = \{\tilde{\pi}(\boldsymbol{\xi}) \in Y\ \forall \boldsymbol{\xi}\}$ as the set of all functions $\tilde{\pi}$ that produce feasible $\boldsymbol{y}$ solutions for all possible random draws $\boldsymbol{\xi}$. Then, $Q_A^* = \max_{\tilde{\pi} \in \Pi} E_{\boldsymbol{\xi}}[Q_A(\boldsymbol{z}(\tilde{\pi}(\boldsymbol{\xi}))]$, where the value of the aggregate quality objective for the given impression arrival sequence is optimized in expectation over all random choices $\boldsymbol{\xi}$ the policy makes, and an optimal randomized policy is a solution to $\tilde{\pi}^* = \arg\max_{\tilde{\pi} \in \Pi}\{E_{\boldsymbol{\xi}}[Q_D(\tilde{\pi}(\boldsymbol{\xi}))] : E_{\boldsymbol{\xi}}[Q_A(\boldsymbol{z}(\tilde{\pi}(\boldsymbol{\xi}))) = Q_A^*\}$.

One of the simplest aggregate quality objectives corresponds to minimizing the cost of *under-delivery*, i.e., the cost incurred by the publisher for exposing fewer individual users than advertisers requested. Using $u_k = (r_k - \sum_j z_{jk})^+$ to denote the under-delivery (i.e., reach shortfall) for campaign k, where $\mathsf{x}^+ = \max(\mathsf{x}, 0)$, we credit shortfalls to the advertiser at the make-good cost rate c_k. Consequently, we can minimize the cost of under-delivery $\sum_k c_k u_k$ by equivalently maximizing the aggregate quality objective $Q_A(\boldsymbol{z}) = -\sum_k c_k (r_k - \sum_j z_{jk})^+$. Because publishers commonly treat revenue from guaranteed ads as booked in advance, it is quite natural to maximize retained revenue (i.e., total revenue collected minus any adjustments due to under-delivery costs), which is equivalent to this aggregate quality objective up to the addition of a scalar constant.

More complex aggregate quality objectives are often used in practice, e.g., to maximize the extent to which exposures are well-spread across the individual users which comprise an advertiser's target market. As described more fully in Ghosh et al. (2009), advertisers prefer when ads are served in a *representative* manner; strictly speaking, this means all users j within ad-

vertiser k's target market $\tilde{\Gamma}(k)$ share the same probability θ_k of being reached, where $\theta_k = r_k/|\tilde{\Gamma}(k)|$. Perfect representativeness is generally difficult to achieve; consequently, we follow Ghosh et al. (2009) and propose minimizing the L2 distance from the perfectly-representative solution. Formally, non-representativeness of campaign k may be defined as $\frac{1}{2\theta_k}\sum_{j\in\tilde{\Gamma}(k)}(E_{\boldsymbol{\xi}}[z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))] - \theta_k)^2$, where the scaling factor $1/(2\theta_k)$ is for mathematical convenience and balancing the relative magnitude of multiple non-representative terms in the full objective. Notice that non-representativeness compares the probability that an individual j is reached by campaign k under randomized policy $\tilde{\pi}$ with the target probability θ_k. By construction, the policy $\tilde{\pi}$ that minimizes non-representativeness is generally not deterministic (e.g., if $\theta_k = 0.5$ and there are only two users $j \in \tilde{\Gamma}(k) = \{1, 2\}$ that match the campaign's targeting, then producing $\{z_{1k} = 1,\ z_{2k} = 0\}$ half the time and $\{z_{1k} = 0,\ z_{2k} = 1\}$ the other half of the time constitutes a randomized policy with $E_{\boldsymbol{\xi}}[z_{1k}] = E_{\boldsymbol{\xi}}[z_{2k}] = \theta_k = 0.5$ and non-representativeness of 0; in contrast, it is easy to verify that any non-randomized policy yields non-representativeness of 1). The aggregate quality objective $Q_A(\boldsymbol{z}) = -\sum_k \frac{w_k}{2\theta_k}\sum_{j\in\tilde{\Gamma}(k)}(E_{\boldsymbol{\xi}}[z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))] - \theta_k)^2 - \sum_k c_k(r_k - \sum_j z_{jk}(\tilde{\pi}(\boldsymbol{\xi})))^+$, when maximized using campaign-specific weights w_k, can be used to produce solutions that have both low under-delivery and low non-representativeness. We use this form of aggregate quality objective in our model, which is equivalent to that of Bharadwaj et al. (2012).

For the disaggregate quality objective $Q_D(\boldsymbol{y})$, there are a number of good candidates publishers can use to ensure each individual user sees ads that (i) are either well-paced over time or purposely delivered successively in a blitz, (ii) are diverse, and/or (iii) do not have competing brands shown to the same user. Although desired, disaggregate quality is typically not explicitly managed by existing ad serving systems. We consider a number of disaggregate quality objectives, which we formally define later. In some cases, it is possible to write $Q_D(\boldsymbol{y})$ as a linear function of the z_{jk} variables. For example, diversity can be measured using $Q_D(\boldsymbol{y}) = \sum_{j,k} z_{jk}$, where $\sum_k z_{jk}$ is the number of distinct ad campaigns shown to a user j. As we will show, such disaggregate quality objectives are particularly convenient since they are computationally easy to optimize.

3.1 Problem Variants

In the general setting just presented, we purposely refrained from characterizing the uncertainty of the impression arrival process. Indeed, one could define a number of problem variants consistent with the above general setting by formalizing different characterizations of the arrival process. More specifically, define Ω as the set of all possible impression arrival sequences that may arise; i.e., for each arrival sequence $\omega \in \Omega$, the total number of arrivals $A(\omega)$, as well as the times and users corresponding to each arrival $(t_1(\omega), j_1(\omega)), (t_2(\omega), j_2(\omega)), \ldots, (t_{A(\omega)}(\omega), j_{A(\omega)}(\omega))$ are dependent on ω. There are a number of assumptions that can be made regarding what is known about (i) the set of instances that we may encounter (i.e., do we know anything about the set Ω that ω will be drawn from, or the probability associated with drawing a specific ω from Ω?), and (ii) once given an instance ω, to what extent do we initially observe ω or some partial information about ω?

At one end of this spectrum, *fully-online* problems consider the input sequence ω to be chosen

fully adversarially (e.g., Mehta et al. 2007). In such settings, the set of possible instances Ω is as broad as the problem description allows. An adversary picks the worst-case $\omega \in \Omega$, and we are not given any initial information about what ω was picked. Moreover, since we do not have a stochastic characterization of Ω to work with, we cannot hope to learn any structure of ω as we observe arrivals over time. At the other end of this spectrum, deterministic or *fully-offline* problems consider the input sequence ω to be fully specified in advance. In such settings, knowing the set of possible instances Ω and a probability distribution over Ω is only useful for characterizing the worst-case or average performance of a particular policy over all possible instances $\omega \in \Omega$. In between fully-online and fully-offline, there are a number of other characterizations of the input sequence. Mehta (2012) is a good reference that describes the differences between a number of so-called input models which have been studied in the context of online bipartite matching and impression-based ad allocation problems. These include the *random permutation* input model (in which the adversary picks $\omega \in \Omega$, which is then randomly permuted and no information about ω is initially provided; see Goel and Mehta 2008; Devanur and Hayes 2009; Feldman et al. 2010; Agrawal et al. 2014), the *Unknown-IID* input model (in which each arrival a's user j_a is drawn independently from an unknown stationary distribution of arrival types, with no information about ω or its distribution over Ω initially provided, but algorithms may learn this distribution over time; see Devanur et al. 2011), and the *Known-IID* input model (which is the same as Unknown-IID except the distribution of arrival types is known in advance, so the algorithm does not need to learn it; see Feldman et al. 2009; Manshadi et al. 2012). Perhaps the closest input model to that of ours is that of Vee et al. (2010) who study an impression-based bipartite matching problem they call *online assignment with forecast*. They endow their algorithm with the ability to obtain a sample of the set of impression types that will arrive online (i.e., a sample of the future). Using our notation, in their input model $\omega \in \Omega$ is adversarially chosen, and then some information about the specific ω that will arrive is revealed to the algorithm. More specifically, a subset of users which will arrive, but not the entire sequence, is revealed. We follow their approach in a number of aspects; in particular, their emphasis on having access to coarse point forecasts and solving a deterministic optimization problem. However, we note that their algorithm is entirely impression-based, and is not suited for serving R&F campaigns. Finally, we note that although the aforementioned input models are often useful constructs for theoretically analyzing the worst-case optimality gap consistent with a given input model, they are all in some way abstractions of reality (e.g., none of them allow for any kind of nonstationarity or forecast errors). Indeed, there are a number of open questions that remain to be settled about how to best model and forecast impression arrival sequences in practice, especially in the context of R&F ads which additionally need to keep track of the specific users that re-visit over the time horizon.

3.2 Model Overview

Our proposed method, which we name Pattern-based Hierarchical Column Generation (Pattern-HCG), was designed to cope with many practical issues. For input, we provide it with only coarse

point forecasts of the impression arrival process. This makes it particularly attractive to use in practice, since in the simplest use case, historical web logs can be sampled to directly provide the necessary characterization of the impression arrival process. More specifically, Pattern-HCG requires a point forecast s_{vi} of the number of users that will arrive over the planning period that are members of demographic i and have browsing behavior v, as well as conservative estimates L_v of the number of times users with browsing behavior v are expected to arrive. Even though Pattern-HCG uses only deterministic inputs, we show it is robust to misspecifications in these forecasts. Most importantly, the set of viewer types indexed by (v, i) does not need to cover all users j that will arrive; it is sufficient to explicitly forecast only the larger, easier-to-forecast ones. Our computational results validate the practicality of our approach out-of-sample on real data, and indicate that our solutions are robust to forecast errors. Furthermore, in Appendix D we extend our deterministic model to explicitly incorporate stochasticity of the number of visits made by each user.

Our Pattern-HCG method first solves an offline deterministic optimization problem, and then uses a robust online phase to process the impression arrival sequence and assign ads to impressions as they arrive. This is consistent with how the best-known online algorithms for non-adversarial input work (see references in §3.1). At the beginning of the planning period, we provide Pattern-HCG with a set of ad campaigns $\mathcal{K}$, a set of demographics $\mathcal{I}$, and a set of viewers' browsing types $\mathcal{V}$, as well as reach and frequency targets (r_k, f_k) for each campaign $k \in \mathcal{K}$, under-delivery penalty rates c_k and non-representativeness weights w_k for each campaign $k \in \mathcal{K}$, point forecasts s_{vi} for the number of users of each demographic-and-browsing type (v, i), sets $\Gamma(k)$ of viewer types (v, i) that each campaign k targets, and conservative estimates L_v of arrival counts for users of each browsing type v. Using this data, we solve a complex bi-criteria optimization problem using a novel iterative procedure with three inter-related components. This produces a plan which tells us (1) the proportion x_{vik} of users in each viewer type (v, i) that should be reached by campaign k, (2) a set of *patterns* $\mathcal{P}_{vi}$ for each viewer type (v, i) such that each pattern defines an exact pre-generated sequence of ads that may be presented to a user, and (3) the number of times y_{vip} that users of type (v, i) should be assigned pattern $p \in \mathcal{P}_{vi}$. (Since we have already used y_{ak} as a variable in this section, we will superscript the vector form of y_{vip} with P to denote "pattern," i.e., $\boldsymbol{y}^P = \{y_{vip}\}$, to distinguish it from $\boldsymbol{y} = \{y_{ak}\}$. Rest assured that there will be no ambiguity later on, since the use of y_{ak} is restricted to this section of the paper). In our online phase, we assign patterns to specific users, randomly drawing patterns at rates consistent with $\boldsymbol{x}$ and $\boldsymbol{y}^P$. Our online phase is also able to construct near-optimal patterns on the fly for users that arrive and cannot be classified as any of the viewer types (v, i) that we explicitly forecasted and optimized for during the offline phase.

Using the main decision variables $\boldsymbol{x}$ and $\boldsymbol{y}^P$ of our model, we can now re-cast the aggregate and disaggregate quality objectives as functions of $\boldsymbol{x}$ and $\boldsymbol{y}^P$. Aggregate quality $\tilde{Q}_A(\boldsymbol{x})$ remains a function of who is reached (in expectation) $\boldsymbol{x}$, and disaggregate quality $\tilde{Q}_D(\boldsymbol{y}^P)$ remains a function of the specific sequence of ads assigned to each user, now represented by $\boldsymbol{y}^P$. Using X to represent the set of all $(\boldsymbol{x}, \boldsymbol{y}^P, \boldsymbol{\mathcal{P}})$ which are mutually consistent and feasible (to be defined more rigorously

later), and defining $\tilde{Q}_A^* = \max_{(\boldsymbol{x},\boldsymbol{y}^P,\boldsymbol{\mathcal{P}})\in X} \tilde{Q}_A(\boldsymbol{x})$, then the bi-objective optimization problem that we seek to solve can be written abstractly as $\max_{(\boldsymbol{x},\boldsymbol{y}^P,\boldsymbol{\mathcal{P}})\in X}\{\tilde{Q}_D(\boldsymbol{y}^P) : \tilde{Q}_A(\boldsymbol{x}) = \tilde{Q}_A^*\}$. To solve this bi-objective problem, we use a tailored optimization method that is in line with the spirit of preemptive (lexicographic) goal programming (see Jones and Tamiz 2010).

Within the context of Pattern-HCG, we can further simplify the general forms of the non-representativeness and under-delivery measures we presented in Section 3, and re-interpret them in this more specific context. Most importantly, by aggregating users into audience segments and explicitly determining the rate at which users within an audience segment are reached, the non-representativeness objective is represented in such a way that we no longer need to consider expectations over multiple replications of the policy to measure it. Treating the proportion x_{vik} as the *probability* that each user j within viewer type (v,i) should be reached, which can be done without loss of optimality, yields $E_{\boldsymbol{\xi}}[z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))] = x_{vik}$. Consequently, non-representativeness of campaign k simplifies to $\frac{1}{2\theta_k}\sum_{j\in\tilde{\Gamma}(k)}(E_{\boldsymbol{\xi}}[z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))] - \theta_k)^2 = \frac{1}{2\theta_k}\sum_{(v,i)\in\Gamma(k)} s_{vi}(x_{vik}-\theta_k)^2$. Written this way, non-representativeness measures how well exposures are spread across different targeted *viewer types* (v,i). By requesting a representative allocation, an advertiser ensures the publisher does not fulfill their entire campaign using some obscure, potentially easy-to-serve subgroup of targeted users. Indeed, if an advertiser targets all users in the USA, they don't expect to only get users in California. Thus, this non-representativeness objective spreads ads across all viewer types (v,i) the advertiser chooses to target, yet makes sure that larger viewer types receive proportionally more ads than smaller ones.

Next, to simplify our under-delivery measure, recall that under-delivery of campaign k was previously defined as $(r_k - \sum_j z_{jk}(\tilde{\pi}(\boldsymbol{\xi})))^+$, a random quantity that depends on the random draws of the policy. We approximate the number of users that campaign k reaches, $\sum_j z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))$, with $E_{\boldsymbol{\xi}}[\sum_j z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))] = \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}$. This approximation is very good since, by the law of large numbers, the sum of a large number of random draws approaches its mean, and the number of random draws made is typically very large (e.g., in the millions)[2]. Consequently, the aggregate quality objective we use in Pattern-HCG has both non-representativeness and under-delivery components and is formally defined as $\tilde{Q}_A(\boldsymbol{x}) = -\sum_k \frac{w_k}{2\theta_k}\sum_{(v,i)\in\Gamma(k)} s_{vi}(x_{vik}-\theta_k)^2 - \sum_k c_k(r_k - \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik})^+$. Finally, we note that formally speaking, similar law-of-large-numbers approximations apply to our disaggregate quality metrics, $\tilde{Q}_D(\boldsymbol{y}^P)$, as well.

The offline phase of our Pattern-HCG method involves a novel pattern-based optimization scheme which iterates between three components: (1) an aggregate reach planning problem which aims to maximize aggregate quality, (2) a pattern assignment problem that maximizes disaggregate quality by assigning patterns to user types in such a way that aggregate quality is maintained, and (3) a pattern generation problem which sequences ads into new patterns for the pattern assignment problem to use.

The aggregate reach planning component of Pattern-HCG is modeled after the formulation of Bharadwaj et al. (2012). Their model involves impression-based campaigns that do not differentiate between 1 person seeing 2 ads vs. 2 people seeing 1 ad each, and therefore cannot directly plan

R&F campaigns. However, the structure of their formulation leads to two very important practical properties which we retain in our model. First, their quadratic non-representativeness penalty function, in conjunction with the specific constraints in our formulation, creates a closed-form relationship between the primal and dual solutions of the reach planning problem. This property is known as *generalizability* (see Vee et al. 2010). Generalizability is important when there are a large number of demographics, and only the most important subset of demographics (e.g., those with enough historical data to accurately forecast) are used to produce the optimal ad allocation. If an arriving user belongs to a demographic that was not explicitly used to construct the optimal ad allocation, then we still can allocate ads near-optimally to this user using the generalizability property. In ad planning models where the aggregate quality metric is linear (e.g., only revenue maximization is considered), this unique mapping between dual and primal solutions does not exist, and the allocation plan is not generalizable. Second, by following the structure of their formulation, we are able to exploit a fast parallelizable primal-dual algorithm developed by those authors called SHALE, which we have adapted to our model and repeatedly call as a subroutine throughout our Pattern-HCG method. We note that different functional forms for the aggregate quality metric (e.g., linear) can be adopted in our framework; however, one would give up both the generalizability property and the ability to use SHALE as an efficient method for solving the aggregate reach planning component of Pattern-HCG. This would be acceptable, for example, if there are only a small number of demographics, since in that case one need not worry about generalizablity and the reach planning math program would be small enough to solve using a commercial solver on a single machine without SHALE.

The remainder of this section lays the foundation for our Pattern-HCG model from the ground up, and is organized as follows. First, in §3.3 we describe the model of Bharadwaj et al. (2012), on which the aggregate reach planning component of Pattern-HCG is based, as well as our basic notation. Then, in §3.4 we show how a heuristic used in practice, called *frequency capping*, may be used to deliver R&F ads in conjunction with an impression-based ad planning model such as the one by Bharadwaj et al. (2012), and point out some of the major differences and distinct issues that arise in R&F planning. We then formally introduce patterns in §4, along with a greedy pattern-generating algorithm, and finally put all the pieces together and introduce our Pattern-HCG model and algorithm in §5. A mathematical notation table is provided in Appendix A for quick reference.

3.3 Allocation of Impression-based Ad Campaigns

A typical method to plan and serve impression-based ads has both an *offline phase* for matching forecasted impression supply with advertisers' impression demand, and an *online phase* for assigning specific ads to arriving users in accordance with the offline impression-based plan. The offline optimization problem is re-solved periodically with updated supply forecasts and each campaign's actual progress (see Chen et al. 2012; Yang et al. 2010).

The offline planning phase has at its core a *bipartite graph*. Each advertising campaign is modeled as a *demand node*, indexed by $k \in \mathcal{K}$, and the publisher's traffic (measured by impressions)

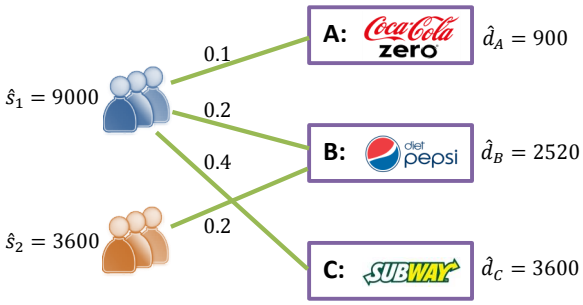


Figure 1: Example Bipartite Graph with Impression-based Ad Campaigns

is partitioned based on user characteristics such as age and gender, geographical location, and behavioral attributes, into *supply nodes*, indexed by $i \in \mathcal{I}$. Figure 1 shows an example with 2 supply nodes and 3 advertising campaigns. The *arcs* model the targeting criteria, i.e., which user types can be served with ads from which campaigns. Letting $\mathcal{T} \subseteq \mathcal{I} \times \mathcal{K}$ denote the set of arcs, we use $\hat{\Gamma}(k) = \{i : (i,k) \in \mathcal{T}\}$ to denote the set of all user types targeted by (eligible for) campaign k, and $\hat{\Gamma}(i) = \{k : (i,k) \in \mathcal{T}\}$ to denote the set of all campaigns that target (can be delivered to) type-i users. Each supply node i represents $\hat{s}_i$ impressions and each campaign k demands a total of $\hat{d}_k$ impressions. We further define $\hat{S}_k = \sum_{i \in \hat{\Gamma}(k)} \hat{s}_i$ as the total volume of impressions that satisfy the targeting criteria of campaign k. The problem is then to find the optimal fraction of impressions from each supply node i that should be allocated to each campaign $k \in \hat{\Gamma}(i)$, denoted $\hat{x}_{ik}$, so as to maximize the quality (or analogously, minimize the cost) of the allocation. Such an optimization problem is known as a *transportation problem* in the operations research literature. Throughout the paper we use the caret ($\hat{}$) to differentiate between quantities that we measure in impressions, as opposed to their analogs (without caret) which we measure as a number of unique users.

The model of Bharadwaj et al. (2012), shown next, plans impression-based guaranteed ads using a transportation formulation with a quadratic objective that minimizes both under-delivery and non-representativeness. We will refer to this as the Impression Allocation (IA) problem:

$$\text{(IA):} \quad \text{Minimize:} \quad \sum_{k, i \in \hat{\Gamma}(k)} \frac{\hat{s}_i}{2\hat{\theta}_k} \hat{w}_k \left(\hat{x}_{ik} - \hat{\theta}_k\right)^2 + \sum_k \hat{c}_k \hat{u}_k \tag{1a}$$

$$\text{s.t.} \quad \sum_{i \in \hat{\Gamma}(k)} \hat{s}_i \hat{x}_{ik} + \hat{u}_k \geq \hat{d}_k \qquad \forall k \tag{1b}$$

$$\sum_{k \in \hat{\Gamma}(i)} \hat{x}_{ik} \leq 1 \qquad \forall i \tag{1c}$$

$$\hat{x}_{ik},\ \hat{u}_k \geq 0 \qquad \forall i, k \tag{1d}$$

Demand constraint (1b) states that the total number of impressions allocated to each campaign k must either exceed its demand $\hat{d}_k$, or otherwise the slack variables $\hat{u}_k$ capture the magnitude of the impression shortfall, called *under-delivery*. Supply constraint (1c) states we cannot allocate more than 100% of supply from each node i. The objective function (1a) penalizes non-representativeness and under-delivery. Each campaign has an under-delivery cost of c_k per impression, and a weight

$\hat{w}_k$ for the importance of achieving a representative allocation. In this impression-based model, a perfectly-representative allocation is one that distributes the demanded impressions of every campaign uniformly across its total eligible supply; i.e., for each impression arrival eligible for campaign k, it assigns the impression to campaign k with probability $\hat{\theta}_k = \hat{d}_k/\hat{S}_k$. As before, non-representativeness quadratically penalizes the deviations from this ideal, and the weights $\hat{s}_i/2\hat{\theta}_k$ are for mathematical convenience and to balance the relative magnitude of each term in the objective. Finally, note that $\sum_i \hat{s}_i - \sum_k (\hat{d}_k - \hat{u}_k)$ impressions will not be allocated to any guaranteed campaign. Although not explicitly modeled here, these *excess* impressions may still get matched to lower-priced non-guaranteed ads in a secondary channel that operates as a spot market to clear excess impressions.

At ad-serving time (i.e., online phase), the optimal solution from (IA) is used as follows: Upon a visit of a type-i user, we randomly draw an eligible ad $k \in \hat{\Gamma}(i)$ with probability $\hat{x}^*_{ik}$. For example, Figure 1 illustrates a 3-campaign 2-demographic example where the numerical solution $\hat{x}^*_{ik}$ is shown on the arcs. Upon a visit from a type-1 user, we draw campaign A (Coca-Cola) with probability $\hat{x}^*_{1A} = 0.1$, campaign B (Pepsi) with probability $\hat{x}^*_{1B} = 0.2$, and campaign C (Subway) with probability $\hat{x}^*_{1C} = 0.4$. There is a 30% chance we do not draw any guaranteed campaign, in which case we assume the user is served a non-guaranteed ad. More ads will be drawn, with the same probabilities, if the webpage has multiple ad slots, since each ad slot corresponds to one impression. Due to the large traffic volume most online publishers have, this random drawing of ads typically achieves the desired proportions $\hat{x}^*_{ik}$ within a short time, while naturally exposing each user to a variety of ads.

The solution illustrated in Figure 1 satisfies all campaign demands with perfect representativeness. Note campaign B (Pepsi) is uniformly spread over the two targeted demographics 1 and 2 as it grabs 20% of each. This translates into $(0.2)(9000) = 1800$ impressions of the larger demographic 1, and $(0.2)(3600) = 720$ impressions of the smaller demographic 2. In other words, campaign B receives 2.5 times more impressions from demographic 1, as it is 2.5 times larger than demographic 2. A total of $(0.3)(9000) + (0.8)(3600) = 5580$ impressions are left unallocated as excess.

The structure of (IA) admits the generalizability property, making it possible to optimize (IA) using only a subset of the largest supply nodes, while still allowing us to recover a near-optimal value for any decision variable $\hat{x}_{ik}$ corresponding to a supply node i that was not explicitly present when (IA) was solved. Specifically, Bharadwaj et al. (2012) show that the primal solution to (IA) can be written as a function of the dual variables of the supply ($\hat{\beta}_i$) and demand ($\hat{\alpha}_k$) constraints in closed-form: $\hat{x}^*_{ik} = \max\{0, \hat{\theta}_k(1 + (\hat{\alpha}^*_k - \hat{\beta}^*_i)/\hat{w}_k)\}$. Moreover, the supply duals ($\hat{\beta}^*_i$) themselves can be calculated directly from the demand duals $\{\hat{\alpha}^*_k$ for $k \in \hat{\Gamma}(i)\}$ without referring to the supply forecast $\hat{s}_i$. Therefore, one only needs to have the vector of optimal demand duals, $\hat{\alpha}^*_k$ (i.e., a single value for each campaign) to be able to reconstruct the optimal primal solution, $\hat{x}^*_{ik}$, in real-time during the serving period. This means that if a type-i user arrives and the supply node i was excluded from (IA) when it was solved, we can use the $\hat{\alpha}^*_k$ values of the campaigns that target this type-i user to determine corresponding near-optimal $\hat{x}_{ik}$ values.

Algorithm 1 Frequency Capping Heuristic (***FreqCap***)

- **OFFLINE:** Solve the impression allocation problem (IA) using $\hat{d}_k = f_k r_k$ as the demand parameters.
- **ONLINE:** Upon a visit from user j from demographic i:
  - If it is the first visit from user j in the planning period: Initialize $q_{jk} = 0$ for all $k \in \hat{\Gamma}(i)$, where q_{jk} counts the number of times user j has been exposed to campaign k.
  - Among the campaigns that target this user $k \in \hat{\Gamma}(i)$ and have not reached their target frequency ($q_{jk} < f_k$): Randomly draw an eligible ad according to implicit probabilities $\hat{x}^*_{ik}$.
  - Increment the frequency counter for the selected campaign k': $q_{jk'} \leftarrow q_{jk'} + 1$.

For a major online publisher with many campaigns and user types, (IA) can easily have hundreds of millions of decision variables. Therefore, using a specialized efficient algorithm to solve (IA) can be crucial. Bharadwaj et al. (2012) develop such an algorithm, called SHALE, that iterates over the dual variables $\hat{\alpha}_k$ and $\hat{\beta}_i$ and converges asymptotically to the optimal dual solution. In §5.1, we extend SHALE to solve the aggregate reach planning component of our R&F allocation problem.

3.4 Frequency Capping

Within the context of delivering impression-based ad campaigns, many publishers use a concept called frequency capping to limit the number of impressions each individual user sees of a given ad. The idea is straightforward. Each campaign k is assigned a maximum frequency $\overline{f}_k$, and the solution to (IA) is used to serve ads in the same manner as described in the previous section, with one small modification. Once a user j of type i sees $\overline{f}_k$ impressions of ad k, then $\hat{x}^*_{ik}$ is treated as if it is zero; i.e., no additional ads of campaign k are shown to this user. Frequency capping prevents any single user from being dramatically over-exposed to an ad just because they happen to spend a lot of time on the publisher's website. As well, frequency caps tend to increase reach, since the publisher must use impressions from a larger group of individuals to satisfy the impression demands $\hat{d}_k$. Within the ad planning literature, we note that Chandler-Pepelnjak and Song (2003) discuss how a campaign's historical performance can be used to find the most efficient or most profitable frequency cap. As well, Buchbinder et al. (2011) develop online algorithms for the publisher to serve impression-based campaigns with minimal under-delivery in the presence of frequency caps.

Because frequency capping is already an existing feature in many impression-based ad-serving systems, it makes a good benchmark to test whether this level of control is sufficient to capably deliver R&F campaigns. In essence, we may consider frequency capping the status quo baseline, with any improvements made in delivering R&F campaigns measured above this baseline. Delivering R&F campaigns using (IA) and a frequency capping heurisitic is accomplished by converting the reach and frequency requirements into total impression demands using $\hat{d}_k = f_k r_k$, and then treating f_k as a frequency cap. The method is formally defined in Algorithm 1.

Despite the apparent similarities that frequency capping has to our problem, note that our frequency requirements, f_k, define the *minimum* number of exposures required for the publisher to

receive payment from an advertiser, whereas a frequency cap, as implemented in current practice, defines the *maximum* number of exposures beyond which the publisher will no longer receive a payment. Indeed, our numerical experiments in §6 show that using frequency capping for serving R&F campaigns causes a significant portion of traffic to be *wasted*, i.e., assigned to users that do not hit the minimum frequency requirement, in which case served impressions are non-billable. This not only leads to considerable under-delivery, but also results in a substantial loss of revenue for the publisher: had the publisher known that the frequency target would not be attained, s/he would have preferred to serve those arrivals with non-guaranteed ads or other R&F campaigns that could reach their frequency target.

We now point out an important distinction between *waste* and *excess*. In the allocation of impression-based ad campaigns, waste does not exist. Each impression is either allocated to a guaranteed campaign and is billable, or is considered excess and served to a non-guaranteed campaign. In either case, the impression generates some revenue. But in the case of allocating R&F campaigns, an impression served to campaign k may either result in a payment (if later that particular user sees the campaign the required f_k times), or is wasted without payment. When the number of visits made by each user is random, any allocation policy is prone to some waste. But to allocate R&F ads well, we should expect that a good policy will need to keep waste in check. As we will see shortly, by clustering users based on their browsing behavior and explicitly planning the sequence of ads that a user sees on successive arrivals, using patterns of a well-chosen length, we can achieve very low waste.

4 Serving Ads using Patterns

We define a serving *pattern* as a sequence of ads arranged over a fixed number of slots, where each slot corresponds to a single ad shown to a user. A particular campaign may appear in multiple slots in a pattern, and a pattern may not necessarily contain all campaigns. Any unassigned slots are treated as excess impressions and may be used to serve non-guaranteed ads. At serving time, when an individual arrives for the first time in the planning period, s/he is assigned a particular pattern. Upon subsequent visits, the ℓ^{th} arrival of the user will be served using the ad in the ℓ^{th} slot of his/her assigned pattern. Arrivals of a user beyond his/her assigned pattern's length are also considered excess and may be served non-guaranteed ads. For ease of exposition, we assume the publisher's webpage has a single ad position. That is, the pattern plans for a single impression upon each arrival, and therefore can be expressed as a one-dimensional array. For an extension to two-dimensional patterns which model multiple impressions per user arrival, see Appendix C.

In addition to keeping waste in check and making it easier to control under-delivery and representativeness of R&F campaigns (i.e., aggregate quality), using explicit patterns also allows the publisher to control disaggregate quality (i.e., user-level pacing, diversity of ads, competition constraints). Figure 2 illustrates a few examples of patterns composed of three guaranteed campaigns {A,B,C}. All patterns are of length 8. In the first two patterns, campaign C appears twice

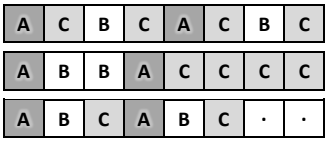


Figure 2: Examples of patterns with three campaigns {A,B,C}

as often as campaigns A or B. The first pattern illustrates uniform pacing (assuming arrivals are also uniform over time), whereas the second pattern delivers campaigns B and C upon successive arrivals, e.g., to strengthen user recall. The last pattern spreads 2 impressions of each campaign uniformly throughout the first 6 slots and leaves the last two slots as excess.

To serve ads using patterns, the publisher should be able to forecast the number of visits that they will get from each user, so a pattern of appropriate length can be constructed for him/her. Assume users are classified according to their browsing behavior, such that all users of the same visit type, $v \in \mathcal{V}$, share a common probability distribution, $\phi_v(\ell)$, that gives the probability of such a user making exactly ℓ visits over the serving period. We can then say that each user of type v will make at least $L_v(\varepsilon) = \Phi_v^{-1}(\varepsilon)$ visits with probability $1 - \varepsilon$, where Φ denotes the CDF of ϕ. With a reasonably small ε, we can use the resulting $L_v(\varepsilon)$ (henceforth referred to in short as L_v) as the anticipated number of visits, and thus an appropriate pattern length, for any user of type v.

Although we take a deterministic modeling approach and henceforth assume that a type-v user makes exactly L_v visits and sees the entire pattern assigned to him/her, our computational experiments in §6 on real industry data show that our solutions are robust to forecast errors and randomness in user arrivals when L_v is chosen as described. For completeness, we present an extension of our model in Appendix D that explicitly takes into account randomness in user arrivals, i.e., the probability distribution $\phi_v(\cdot)$, when sequencing ads into patterns. As can be expected, a probabilistic model takes longer to solve than a deterministic one.

Patterns can either be generated on the fly as-needed, or pre-generated in advance. The greedy pattern-based method we introduce in the subsequent section shows how we can generate patterns on the fly using the solution to a reach-based variant of the Impression Allocation problem (IA). Afterward, in §5 we will show how we pre-generate and then serve optimal patterns using our Pattern-HCG method.

4.1 Reach-and-Frequency Ad Allocation Using Greedily-Constructed Patterns

Recall from §3.3 that to plan and serve impression-based ads, we first solved a math program to match the supply of impressions with the demand of impressions (offline phase), and then used the resulting optimal allocation to serve ads to users upon arrival in real-time (online phase). Our greedy pattern-based method also has offline and online phases.

In the offline phase, we solve a variation of (IA) which we call the Reach Allocation problem (RA). The math program (RA) differs from (IA) in three main aspects. First, the ad allocation is represented by unique individuals, rather than impressions. Second, supply nodes partition users by

both demographic and predicted number of visits, rather than only demographic. Third, the supply constraints become more complex, to model the relationship between individuals and impressions.

To formally define (RA) we need some additional notation. Noting that campaigns requiring a frequency of f_k can only be assigned to users that visit at least f_k times, we define our eligible matching sets as $\Gamma(k) = \{(v,i) : (i,k) \in \mathcal{T}, L_v \geq f_k\}$ and $\Gamma(v,i) = \{k : (i,k) \in \mathcal{T}, f_k \leq L_v\}$. Let s_{vi} denote the number of unique users of visit type v within demographic i that will arrive over the planning horizon, and let $S_k = \sum_{(v,i)\in\Gamma(k)} s_{vi}$ denote the total number of unique users that satisfy the targeting criteria of campaign k. For a perfectly representative allocation, each campaign k should grab a $\theta_k = r_k/S_k$ proportion of type-$(v,i) \in \Gamma(k)$ users. Consequently, c_k and w_k are the cost per unit of under-delivery and non-representativeness penalty weight, respectively for campaign k, that apply when under-delivery and representativeness are measured in individuals rather than impressions. Our decision variables are now x_{vik}, which measures the proportion of type-(v,i) users that should be reached by (i.e., exposed to f_k impressions of) campaign k; and u_k, which measures the under-delivery of campaign k (i.e., the shortfall in attaining campaign k's reach target r_k). Our Reach Allocation problem (RA) is as follows:

$$\text{(RA):} \quad \text{Minimize:} \quad \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}}{2\theta_k} w_k \left(x_{vik} - \theta_k\right)^2 + \sum_k c_k u_k \tag{2a}$$

$$\text{s.t.} \quad \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} + u_k \geq r_k \qquad \forall k \tag{2b}$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq 1 \qquad \forall v,i \tag{2c}$$

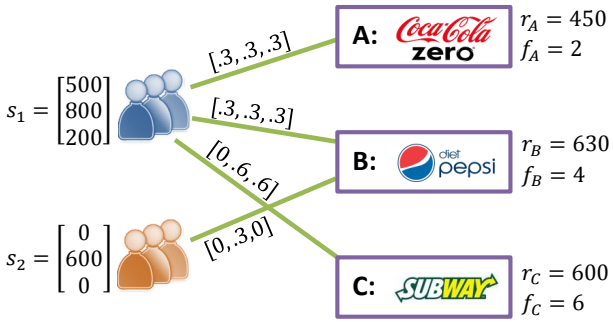
$$0 \leq x_{vik} \leq 1 \qquad \forall v,i,k \in \Gamma(v,i) \tag{2d}$$

$$u_k \geq 0 \qquad \forall k \tag{2e}$$

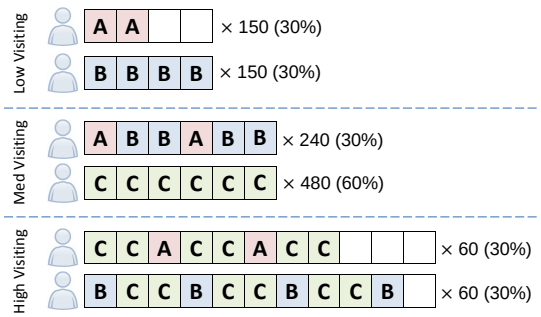
Demand constraint (2b) requires the total number of unique users reached by each campaign k to meet or exceed r_k, or otherwise the slack variables u_k capture the magnitude of under-delivery.

Supply constraint (2c) is structurally different from its counterpart (1c) in (IA). A naïve translation of (1c) yields $\sum_{k\in(v,i)} x_{vik} \leq 1$. However, we can immediately see that such a constraint would be too strict. Indeed, if campaigns A and B each require only one impression (i.e., $f_A = f_B = 1$), and every user of type (v,i) arrives at least twice, then it is possible to reach each individual by both campaigns, i.e., $x_{viA} = x_{viB} = 1$, which violates $\sum_{k\in(v,i)} x_{vik} \leq 1$. Instead, we write the supply constraint in the impression space, and translate users reached into impressions. By multiplying through by $L_v s_{vi}$, the supply constraint (2c) is equivalent to $\sum_{k\in\Gamma(v,i)} f_k s_{vi} x_{vik} \leq s_{vi} L_v$. In this expanded form, the left-hand side represents all impressions allocated from supply node (v,i), where each of the $s_{vi}x_{vik}$ individuals served campaign k are exposed to f_k impressions. The right-hand side reflects the total number of impressions from supply node (v,i) that are available for R&F campaigns, and is computed as the number of individuals s_{vi} of type (v,i), multiplied by the pattern length L_v (measured in impressions) used for this user type. Finally, we note that since (2c) does not imply $x_{vik} \leq 1$ as its counterpart (1c) in (IA) did, we now explicitly enforce the upper-bounds $x_{vik} \leq 1$ using constraint (2d) to ensure x_{vik} can be interpreted as a proportion.

Figure 3 provides a solution to an instance of (RA), as well as one possible extension of this



(a) Bipartite graph, supply of users in {low,med,high}-visiting classes within each demographic, reach and frequency parameters, and the optimal reach allocations obtained by solving (RA).



(b) A pattern-based assignment of ads for demographic 1 that match the optimal reach allocations given by (RA). {Low,Med,High}-visiting users make {4,6,11} visits each, respectively.

Figure 3: Example Bipartite Graph and Pattern-Based Solution of R&F Campaigns

solution to specific patterns. In this example, the publisher receives visits from $s_1 = 1500$ unique individuals of demographic 1, of which $\{500, 800, 200\}$ users are classified as {low,med,high}-visiting, and make $\{4, 6, 11\}$ page visits, respectively, for a total of $\hat{s}_1 = 9000$ impressions. All $s_2 = 600$ users of demographic 2 are med-visiting and make exactly 6 visits each, producing a total of $\hat{s}_2 = 3600$ impressions. Campaigns A, B, and C require $\{450, 630, 600\}$ unique users to see $\{2, 4, 6\}$ impressions, respectively, to be considered reached. Note that the demands and supplies, when translated into impressions (e.g., using $\hat{d}_k = f_k r_k$), match those of our earlier example from Figure 1.

In Figure 3(a), the values on the arcs show the optimal solution x^*_{vik} obtained by solving (RA). This solution satisfies all campaigns' reach requirements and achieves perfect representativeness. Among the $s_1 = 1500$ users of demographic 1, 30% (450 individuals) are reached by campaign A (i.e., each see $f_A = 2$ impressions of the Coca Cola ad), 30% (450 individuals) are reached by campaign B (i.e., each see $f_B = 4$ impressions of the Pepsi ad), and 60% of med- and high-visiting users (600 individuals) are reached by campaign C (i.e., each see $f_C = 6$ impressions of the Subway ad). Note that low-visiting users arrive only 4 times which is not enough to be allocated to campaign C. Finally, among the $s_2 = 600$ users of demographic 2, 30% of med-visiting users (180 individuals) are reached by campaign B.

Figure 3(b) demonstrates one possible pattern-based assignment corresponding to the reach fractions x^*_{vik} within demographic 1. For the 500 low-visiting users who make 4 visits each, we assign 30% (150 individuals) a pattern with only campaign-A impressions, and another 30% (150 individuals) a pattern with only campaign-B impressions. For the 800 med-visiting users who make 6 page visits each, we assign 30% (240 individuals) a pattern with impressions from both campaigns A and B, and 60% (480 individuals) a pattern with only campaign-C impressions. For the 200 high-visiting users who make 11 visits each, we assign 30% (60 individuals) a pattern with campaigns A and C, and 30% (60 individuals) a pattern with campaigns B and C. Note that whenever campaign k is in a pattern, exactly f_k impressions are allotted to campaign k. Finally,

Algorithm 2 Pattern-based Greedy Heuristic (***Pattern-G***)

- **OFFLINE:** Solve the reach allocation problem (RA).
- **ONLINE:** Upon a visit from user j from of type (v, i):
  - If it is the first visit from user j in the planning period: Initialize an empty pattern, $P_j = \{\}$. Follow a random permutation of eligible campaigns $k \in \Gamma(v, i)$ and conduct a Bernoulli experiment with success probability x^*_{vik} to determine whether the user should be reached by each $k \in \Gamma(v, i)$. If campaign k is selected, add f_k impressions of k to the pattern P_j. However, if adding k makes the pattern longer than L_v, instead stop without adding k and store P_j.
  - Randomly draw one impression from P_j to show to the user. Remove that impression from P_j.

$\{200, 80, 80\}$ individuals of {low,med,high}-type are not served any R&F campaign, and all of their page visits are excess impressions. Similarly, all unfilled slots in the illustrated patterns are excess impressions.

Our greedy heuristic, defined in Algorithm 2, uses the solution obtained from (RA) and constructs and assigns a pattern to a user upon his/her first visit. It creates a pattern for a type-(v, i) user by randomly selecting full blocks of f_k impressions from campaigns $k \in \Gamma(v, i)$ according to a Bernoulli process with success probabilities x^*_{vik}, until the L_v slots are full. If the user sees the full pattern, s/he sees exactly f_k impressions required to be counted as reached, and no impressions are wasted. The greedy heuristic does not explicitly optimize disaggregate quality metrics such as user-level pacing or diversity. However, we do pay some attention to disaggregate quality by serving impressions from the pattern in random order; this spreads out each selected campaign's ads and thus provides some amount of user-level pacing. Finally, we note that (RA) maintains enough similarity to (IA) that it is generalizable and we can adapt SHALE to solve it efficiently; we will discuss this further in §5.

Because Pattern-G constructs patterns on-the-fly, its patterns may not make efficient use of all L_v impressions from users of type v. Consequently, although Pattern-G aims to meet the reach fractions x^*_{vik} prescribed by the optimal solution of (RA), it could fall short when the combinatorial problem of packing blocks of f_k impressions into patterns is difficult. In the following section, we introduce a method which explicitly considers the packing problem of pattern generation, and pre-generates optimal patterns.

5 Pattern-based Hierarchical Column Generation

Column generation as developed by Gilmore and Gomory (1961) was designed to solve a single-objective optimization problem known as the *cutting stock problem.* Using notation analogous to our R&F planning problem, in the cutting stock problem a manufacturer must produce r_k strips of length f_k to satisfy the demands of all customers $k \in \mathcal{K}$ by cutting standard-sized length-L pieces of stock material (e.g., rolls of metal or paper) into strips of varying lengths. The objective is either to minimize the number of stock rolls used, or minimize the amount of material scrapped;

when over-production is not an option, these two are equivalent (see Appendix I). Determining how to cut strips from rolls is in general a combinatorially challenging problem. For example, given $L = 10$ with two desired strip lengths $f_A = 3$ and $f_B = 4$, the only pattern with zero scrap is $\{3, 3, 4\}$. Consequently, if demand for 3-unit strips is exactly double that of 4-unit strips, i.e., $r_A = 2r_B$, then we can satisfy the demands without producing any scrap. However, for any other demand levels, some scrap will be produced, and we would need to consider using other patterns, such as $\{3, 3, 3, 1\}$ and $\{4, 4, 2\}$. Column generation is a duality-based technique that tackles the combinatorially challenging problem of implicitly considering all possible ways that patterns can be constructed to decide which patterns to use, and how many times to use each pattern. We use the duality-based constructs from classical column generation to produce patterns for sequencing ads to users. However, our R&F planning problem is more complex than the classical cutting stock problem, and consequently our Pattern-HCG method is also substantially more complex.

We begin this section by highlighting the main structural differences between the cutting stock problem and our R&F ad planning problem. In our context, the set of arrivals from each unique user constitutes a stock roll. However, rather than there being only one type of roll as in the cutting stock problem, we have one roll type for each user type (v, i). Roll length is determined by the anticipated number of visits L_v, while the user's demographic i can be thought of as providing the roll with some other attribute, e.g., its color. Moreover, whereas the cutting stock problem assumes an infinite number of rolls are available, we have s_{vi} forecasted users of type (v, i), which constitutes a fixed capacity for each roll type. Like the cutting stock problem, we aim to produce r_k strips of length f_k, so that r_k users can be exposed to f_k impressions. However, in our case, since each block of f_k impressions assigned to advertiser k must come from a different user, we can only ever cut a strip of type k once from the same roll. In contrast, the cutting stock problem allows multiple strips of type k to be cut from the same roll.

With regards to the objective function, we note that our problem has a primary objective (maximize aggregate quality) and a secondary objective (maximize disaggregate quality). Recall that our proposed aggregate quality metric not only minimizes under-delivery, but also maximizes representativeness. Maximizing representativeness involves spreading impressions across targeted demographics, and is analogous to not only cutting a total of r_k strips of length f_k, but also striving to deliver to the customer a well-balanced mix of different-colored strips, which to the best of our knowledge, has not been considered in the cutting stock literature. Furthermore, most disaggregate quality metrics that apply to R&F planning are different from what is relevant to a cutting stock problem. First, note that what we consider excess is scrap (or trim loss) within the context of the cutting stock problem and there is no corresponding concept of waste. Having excess impressions, especially toward the end of a pattern, can increase the robustness of our solution to uncertainty in the number of arrivals for a given user, and thus reduce waste. Therefore, minimizing excess (equivalent to minimizing scrap or the number of rolls, which are the usual objectives in cutting stock) is not an ideal objective for our R&F planning model. A somewhat less popular objective in cutting stock is to minimize the number of cuts in the patterns (which saves labor and machine

time). In our case, the number of cuts corresponds to the number of campaigns, i.e., the diversity of ads served to a user; which is something we would prefer to maximize instead. Finally, some disaggregate quality metrics require us to model each unit of stock as if they are ordered; for example, to spread impressions to a user over time, we care about the actual sequence and not just the number of times the user is exposed. In contrast, the cutting stock problem's stock units are not ordered in any particular manner. Thus, there are several distinct differences between the standard cutting-stock problem and our more involved R&F ad planning problem.

In Hojjat et al. (2014) we studied a variant of the R&F ad planning problem that is closer in structure to the classical cutting stock problem. In that conference paper, we also had ad campaigns that require r_k users to see f_k impressions, and viewer types (v, i) that correspond to heterogeneous rolls with different lengths and colors. But in contrast to the problem studied in this paper which has both primary and secondary objectives, the problem in Hojjat et al. (2014) had only a single objective, defined as the weighted sum of under-delivery, non-representativeness, and pattern-related costs. For that problem, we proposed a two-step solution procedure modeled after classical column generation, with a master problem for pattern assignment and a related pattern-generating subproblem. Although theoretically correct, the model presented in Hojjat et al. (2014) suffered a number of practical issues. In particular, our master problem in that paper did not retain enough of the structure of (RA) to allow us to uniquely characterize the primal solution as a function of the dual solution (for details, see Appendix E). As a result, the solution was not generalizable, and second, we could not use SHALE as a fast algorithm to solve the master problem. Recall that generalizability is important when dealing with a large number of demographics, and so is having a fast algorithm for solving the large master problem which is solved numerous times in our iterative procedure. Third, the emphasis on a single objective function in Hojjat et al. (2014) meant that every iteration of column generation was focused on improving disaggregate pattern quality, which was computationally expensive. In contrast, by focusing on the aggregate and disaggregate pattern quality objectives at different stages, our Pattern-HCG method spends several iterations first in a faster feasibility-seeking phase, before finishing with an optimality-seeking phase where disaggregate pattern quality is addressed in a distributed parallelizable fashion. Fourth, and lastly, including the disaggregate pattern quality terms in the composite objective of Hojjat et al. (2014) led to a difficult-to-resolve scaling issue. From our experience, applying a low weight to pattern quality resulted in low-quality patterns which did not justify the high computational effort in generating them. And applying a high weight to pattern quality induced high under-delivery and low representativeness, which have a direct revenue consequence for the publisher. Re-casting the problem as one with primary aggregate quality and secondary disaggregate quality objectives alleviates the need to figure out what the appropriate scaling factor is that balances these two competing objectives.

In the following, we introduce our new approach which retains the benefit of generating patterns using column generation, but does not suffer from the four issues just mentioned. We begin by describing the three distinct components of Pattern-HCG: reach allocation, pattern generation, and

pattern assignment. Then, we describe how we coordinate these components in an iterative fashion.

5.1 Reach Allocation

The reach allocation component of Pattern-HCG chooses the proportion of users x_{vik} of each type (v,i) to assign to each campaign k so as to maximize aggregate quality (i.e., minimize non-representativeness and under-delivery). It is modeled by the following quadratic program, which has decision variables x_{vik} and u_k:

$$\text{(RA-}\delta\text{):} \quad \text{Minimize} \quad \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}}{2\theta_k} w_k \left(x_{vik} - \theta_k\right)^2 + \sum_k c_k u_k \qquad \underline{\text{Duals}} \text{ (All} \geq 0) \qquad (3a)$$

$$\text{s.t.} \quad \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} + u_k \geq r_k \qquad \forall k \qquad \alpha_k \qquad (3b)$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq \delta_{vi} \qquad \forall v,i \qquad \beta_{vi} \qquad (3c)$$

$$0 \leq x_{vik} \leq 1 \qquad \forall v,i,k \in \Gamma(v,i) \qquad \gamma^L_{vik}, \gamma^U_{vik} \qquad (3d)$$

$$u_k \geq 0 \qquad \forall k \qquad \varphi_k \qquad (3e)$$

This formulation improves upon our earlier reach allocation problem (RA) by introducing *impression utilization factors* $\delta_{vi} \in [0,1]$ for each supply constraint (v,i). Note that the supply constraint (3c) is a generalization of our earlier supply constraint (2c) from (RA) which assumed $\delta_{vi} = 1$. When $\delta_{vi} = 1$, all $s_{vi}L_v$ impressions of supply node (v,i) are eligible to be assigned to R&F campaigns. But, more generally, $(1-\delta_{vi})\%$ of the impressions from supply node (v,i) are set aside as excess, leaving $\delta_{vi}(s_{vi}L_v)$ eligible for R&F campaigns. As we will see in §5.4, due to the combinatorial difficulty of packing groups of ad exposures into patterns, the patterns we construct often have some inevitable amount of excess (i.e., slots not assigned to any R&F campaign). This corresponds to trim loss or scrap in the cutting stock problem which cannot be avoided unless the size and length of orders allow for a perfect cut from stock rolls. Consequently, the impression utilization factors δ_{vi} are used by our method to control how optimistic or pessimistic (RA-δ) should be in apportioning impressions to campaigns.

We now establish the relationship between the optimal primal and dual solutions of (RA-δ). The proof of the following theorem is based on the Karush-Kuhn-Tucker (KKT) conditions, and is provided in Appendix F.

Theorem 1. *The optimal primal and dual solutions of (RA-δ) satisfy the following relationships:*

1. *The optimal primal solution x^*_{vik} can be computed from the optimal dual solution $\{\alpha^*_k, \beta^*_{vi}\}$, and is given by: $x^*_{vik} = g_{vik}(\alpha^*_k, \beta^*_{vi}) \equiv \min\Big[1, \max\Big[0, \theta_k + \frac{\theta_k}{w_k}\big(\alpha^*_k - \frac{f_k}{L_v}\beta^*_{vi}\big)\Big]\Big]$.*
2. *For each campaign k, we have $\alpha^*_k \in [0, c_k]$. Furthermore, either $\alpha^*_k = c_k$, or the demand constraint binds with no under-delivery, i.e., $\sum_{(v,i)\in\Gamma(k)} s_{vi}x^*_{vik} = r_k$. The optimal solution never over-delivers a campaign.*
3. *For each supply node (v,i), we have $\beta^*_{vi} \in \Big[0,\ \max_{k\in\Gamma(v,i)} \frac{w_k+\alpha^*_k}{f_k} L_v\Big]$. Furthermore, either $\beta^*_{vi} = 0$ or the supply constraint binds, i.e., $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x^*_{vik} = \delta_{vi}$.*

Algorithm 3 The Modified SHALE Algorithm

- INITIALIZE: Set all $\alpha_k = 0$ (or any other value in $[0, c_k]$ that satisfies the assumptions in Theorem 2).
- REPEAT:
  - STEP 1: (Parallelize) For each (v, i), find β_{vi} such that: $\sum_{k \in \Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k, \beta_{vi}) = \delta_{vi}$. Binary search over interval $\left[0,\ \max_{k \in \Gamma(v,i)} \frac{w_k + \alpha_k}{f_k} L_v\right]$. If no solution exists, set $\beta_{vi} = 0$.
  - CHECK: If suitable optimality gap, iteration or time limit is attained, terminate.
  - STEP 2: (Parallelize) For each k, find α_k such that: $\sum_{(v,i) \in \Gamma(k)} s_{vi} g_{vik}(\alpha_k, \beta_{vi}) = r_k$. Binary search over interval $[0, c_k]$. If no solution exists, set $\alpha_k = c_k$.

4. The optimal solution to (RA-δ) is unique.

In Algorithm 3, we generalize the SHALE algorithm of Bharadwaj et al. (2012) and use it to efficiently solve (RA-δ). The algorithm iterates through the dual space, and converges to the solution to the KKT system of (RA-δ). Step 1 attempts to improve β_{vi}, and invokes parts 1 and 3 from the theorem to find the unique value of β_{vi} which satisfies the KKT conditions under the assumption that all α_k's are optimal. Similarly, Step 2 attempts to improve α_k, and invokes parts 1 and 2 of the theorem to find the unique value of α_k which satisfies the KKT conditions under the assumption that all β_{vi}'s are optimal. Overall, SHALE can be viewed as an algorithm which maintains stationarity and dual feasibility throughout, while striving for primal feasibility and complementary slackness. More specifically, primal feasibility always holds immediately following Step 1. If at that point complementary slackness is also attained, then optimality is achieved and the algorithm terminates.

Bharadwaj et al. (2012) provide a proof of convergence for SHALE, and show that the algorithm makes smooth progress towards bucketing campaigns into two groups: those with either zero or non-zero under-delivery at the optimal solution. Specifically, they show that after $\frac{1}{\epsilon}|\mathcal{K}| \max_k\{c_k/w_k\}$ iterations, SHALE produces a primal solution that, for each campaign k, either $\alpha_k = c_k$ (under-delivery is being priced in), or at least $(1 - \epsilon)\%$ of the demand (i.e., reach) r_k is satisfied. We provide a generalized proof of convergence in Appendix G which does not rely on all α_k values being initialized to zero at the start of the algorithm, as in Bharadwaj et al. (2012). This is important for us, since Pattern-HCG solves (RA-δ) multiple times with δ_{vi} values monotonically decreasing at each iteration. Warm-starting using the optimal α_k values from the previous iteration provides significantly faster convergence.

Theorem 2 (Convergence of Modified SHALE)**.** *Given a vector of impression utilization factors δ, the Modified SHALE Algorithm converges to the optimal dual solution for (RA-δ) as long as either (i) all α_k values are initialized to zero, or (ii) we initialize $\alpha_k = \alpha'_k, \forall k \in \mathcal{K}$ where α' is the optimal dual solution to (RA-δ') for which $\delta' \geq \delta$ componentwise.*

Finally, we state how we use Theorem 1 to produce a near-optimal primal solution $x_{v'i'k}$ for a user of type (v', i') which was not explicitly considered as a supply node when (RA-δ) was solved.

Corollary 1 (Generalizability)**.** *For any unexpected user visit of type (v', i'), we can identify the set of targeted campaigns $\Gamma(v', i')$ and use the corresponding $\alpha^*_k \in \Gamma(v', i')$ to estimate $\beta^*_{v'i'}$ using Step 1 of the Modified SHALE Algorithm*[3]*. From part 1 of Theorem 1, a corresponding primal solution is $x_{v'i'k} = g_{v'i'k}(\alpha^*_k, \beta^*_{v'i'})$. Moreover, by construction, the supply constraint is satisfied, hence $\{x_{v'i'k} : k \in \Gamma(v', i')\}$ is feasible.*

Assuming generalized arrivals do not account for a significant portion of the publisher's traffic, the dual solution α^*_k obtained by solving (RA-δ) will be close to the true optimum (i.e., that of (RA-δ) with supply nodes for all generalized arrivals). Therefore, the generalized solution proposed in Corollary 1 is near optimal.

5.2 Pattern Assignment

The pattern assignment component of Pattern-HCG determines how patterns should be assigned to users of each demographic and visit-type to maximize disaggregate quality while ensuring that the pattern assignment is consistent with the reach allocation from (RA-δ). Let $\mathcal{P}_{vi}$ denote the set of all patterns that can be assigned to users of type (v, i). It suffices to initially assume that $\mathcal{P}_{vi}$ contains all patterns of length L_v that can be constructed by picking a subset of campaigns $\mathcal{K}' \subseteq \Gamma(v, i)$ that fit within the pattern (i.e., $\mathcal{K}'$ satisfies $\sum_{k \in \mathcal{K}'} f_k \leq L_v$), and then permuting the $\sum_{k \in \mathcal{K}'} f_k$ impressions from the chosen campaigns into the L_v slots of the pattern. Let π_{vip} be the cost (i.e., lack of disaggregate quality) of pattern $p \in \mathcal{P}_{vi}$, and b_{kp} be a binary parameter that indicates whether or not f_k impressions of campaign k are in pattern p. The following linear program determines the optimal number of times each pattern p should be assigned to type-(v, i) users, denoted y_{vip}, in order to minimize pattern assignment cost (i.e., maximize disaggregate quality):

$$
\begin{aligned}
\text{(PA):} \quad \Psi_{vi} := \text{ Minimize } & \sum_{p \in \mathcal{P}_{vi}} \pi_{vip} y_{vip} && \underline{\text{Duals:}} && \text{(4a)} \\
& \sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip} = s_{vi} x^*_{vik} \quad \forall k \in \Gamma(v, i) && \bar{\alpha}_{vik} \text{ (free)} && \text{(4b)} \\
& \sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq s_{vi} && \bar{\beta}_{vi} \geq 0 && \text{(4c)} \\
& y_{vip} \geq 0 \quad \forall p \in \mathcal{P}_{vi} && - && \text{(4d)}
\end{aligned}
$$

Constraint (4b) ensures the number of type-(v, i) users reached by campaign k equals the number (RA-δ) determined should be reached by campaign k. Since the optimal solution to (RA-δ) is unique (part 4 of Theorem 1), maintaining the aggregate quality attained by (RA-δ) is equivalent to matching each and every variable x^*_{vik}. Constraint (4c) ensures we do not assign more patterns than there are users available (as each user can be assigned at most one pattern). Producing a pattern assignment involves solving one such linear program for each user type (v, i).

The set of all possible patterns for any given user type (v, i) can be exponentially large; thus, solving (PA) involves considering a linear program with an exponential number of variables. The column generation technique allows us to implicitly, rather than explicitly, consider all possible

patterns. The idea stems from the fact that most patterns will not be part of the optimal pattern assignment. For any such pattern p' where $y^*_{vip'} = 0$, we can exclude p' from $\mathcal{P}_{vi}$ and still obtain the same optimal solution. Consequently, we can solve (PA) to optimality by explicitly considering only a small subset of patterns in the pattern pool $\mathcal{P}_{vi}$, as long as the pool contains all patterns that are part of the optimal pattern assignment. Although it would seem like an insurmountable problem to determine a small yet sufficient set of patterns, column generation is an iterative technique that does just that. It begins by initializing the pattern pool $\mathcal{P}_{vi}$ with a small set of patterns that can produce a feasible solution to (PA). Then, at each iteration, a pattern generation problem is solved to identify the patterns which, at the margin, improve the value of the solution; these patterns are added to the pattern pool. This is repeated until no improving pattern exists, at which point (PA) is solved to optimality while the pattern pool $\mathcal{P}_{vi}$ contains many fewer patterns than the explicit set of patterns represented by all combinations of campaigns that fit within a pattern and all permutations of their impressions.

5.3 Pattern Generation

The pattern generation component of HCG is used to produce new patterns. It uses the dual solution from the current pattern assignment to determine, at the margin, what pattern is most beneficial to add to each pattern pool $\mathcal{P}_{vi}$. The reduced cost of the y_{vip} variable in (PA) is given by $\pi_{vip} - \sum_{k\in\Gamma(v,i)} \bar{\alpha}^*_{vik} b_{kp} + \bar{\beta}^*_{vi}$, where $\{\bar{\alpha}^*_{vik}, \bar{\beta}^*_{vi}\}$ is the current dual solution for user type (v,i). Therefore, the pattern generation problem, which constructs a new pattern for user type (v,i), is:

$$\text{(PG):} \qquad \psi_{vi} := \text{Minimize} \quad \pi(\mathbf{b}) \; - \sum_{k\in\Gamma(v,i)} \bar{\alpha}^*_{vik} b_k \tag{5a}$$

$$\text{s.t.} \quad \sum_{k\in\Gamma(v,i)} f_k b_k \le L_v \tag{5b}$$

$$b_k \in \{0,1\}, \quad \forall k \in \Gamma(v,i) \tag{5c}$$

The binary variables b_k, $k \in \Gamma(v,i)$, determine whether or not campaign k is included in the new pattern. Since including k requires f_k slots of the pattern, constraint (5b) ensures the total number of slots used is within the pattern length L_v. For any fixed vector of decisions $\mathbf{b} = (b_k)_{k\in\Gamma(v,i)}$, the function $\pi(\mathbf{b})$ determines the cost (i.e., lack of disaggregate quality) of the new pattern. The second part of the objective, $\sum_{k\in\Gamma(v,i)} \bar{\alpha}^*_{vik} b_k$, is linear in the decision variables b_k. Dual values $\bar{\alpha}^*_{vik}$ computed previously by (PA) are constants here, and measure how important it is to select each campaign $k \in \Gamma(v,i)$ in order to achieve the reach allocation x^*_{vik} of (RA-δ).

The complexity of (PG) depends on the choice of function $\pi(\mathbf{b})$. For any $\pi(\mathbf{b})$ which is linear in the b_k variables, (PG) can be formulated as a binary knapsack problem, which is theoretically NP-hard but admits a Fully Polynomial-Time Approximation Scheme (FPTAS) and can be solved very quickly using dynamic programming in $O(|\Gamma(v,i)|L_v^2)$ time (see Martello and Toth, 1990, Ch.2). Some examples of disaggregate quality metrics that can be implemented using a linear $\pi(\mathbf{b})$ include maximizing the diversity of ads and/or the number of excess slots within a pattern. Using a linear $\pi(\mathbf{b})$ metric, we can solve more than 1000 instances[4] of (PG) per second on a single 2.4GHz CPU.

Another useful disaggregate quality metric is user-level pacing, i.e., how well-spread impressions of the same campaign are over time. But since pacing is a metric that not only depends on the set of campaigns within the pattern, but also on how impressions are sequenced within the pattern, it cannot be implemented using a linear $\pi(\mathbf{b})$. Such a pacing metric $\pi(\mathbf{b})$ involves an inner-optimization problem to uniformly arrange impressions over pattern slots given the set of chosen campaigns $\mathbf{b}$. Using CPLEX, solving an instance of an extended formulation of (PG) that has additional binary variables and constraints to keep track of the specific sequence of impressions within the pattern can take tens of seconds. This is an order of magnitude slower than solving a binary knapsack problem via dynamic programming as we do when $\pi(\mathbf{b})$ is linear, but it is important to note that (PA) and (PG) are solved independently for each supply node (v, i), and thus can be run in parallel across many machines. This slower runtime for each instance of (PG) is still within practical limits given that large publishers in industry have thousands of parallel computing nodes at their disposal. For the explicit functional forms of $\pi(\mathbf{b})$ and the corresponding models for the disaggregate quality metrics concerning (i) diversity of ads served to each user, (ii) optimal amount of excess in the patterns, and (iii) user-level pacing of ads over time, please see Appendix B.

5.4 The Pattern-HCG Algorithm

Pattern-HCG combines the three components of the preceding subsections (reach allocation, pattern assignment, and pattern generation) in an integrated, iterative fashion. At a high level, the idea is to first solve (RA-δ) to produce an aggregate reach allocation with maximum aggregate quality, and then use column generation to generate and assign patterns to maximize disaggregate quality while maintaining the aggregate quality attained by (RA-δ). In the process, there are two substantial challenges that must be overcome. First, we need a way to construct an initial set of patterns so we can start with a feasible solution to (PA). Second, while searching for feasible patterns we may learn that (PA) is infeasible for some user types (v, i). When that happens, we re-solve (RA-δ) with a lower δ_{vi}, and iterate. Consequently, the full Pattern-HCG algorithm has two phases: (1) a feasibility phase in which the focus is on aggregate quality and δ_{vi} values are iteratively tuned to ensure that the solution to (RA-δ) can be translated into a pattern assignment by (PA) for every user type, and (2) a pattern improvement phase which focuses exclusively on optimizing the secondary, disaggregate quality objective without sacrificing the value we obtained for the primary, aggregate quality objective at the end of the feasibility phase.

The feasibility phase begins by initializing the impression utilization factors δ_{vi} to 1 for all user types (v, i). We construct a reach allocation by solving (RA-δ), and then we solve a modified

version of the pattern assignment problem (PA) for each user type (v,i):

$$\text{(PA-F):} \qquad \Psi_{vi}^{(F)} := \text{Minimize} \quad \sum_{p\in\mathcal{P}_{vi}} y_{vip} \qquad \text{Duals:} \qquad (6a)$$

$$\sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip} = s_{vi} x^*_{vik} \quad \forall k \in \Gamma(v,i) \qquad \bar{\alpha}_{vik}^{(F)} \text{ (free)} \qquad (6b)$$

$$y_{vip} \geq 0 \quad \forall p \in \mathcal{P}_{vi} \qquad - \qquad (6c)$$

Since we ignore disaggregate pattern quality in the feasibility phase, the pattern costs π_{vip} of (PA) do not factor into the objective. Instead, we relax the supply constraint (4c) and minimize its left-hand side, i.e., the number of users allocated by this pattern assignment, $\sum_{p\in\mathcal{P}_{vi}} y_{vip}$. Unlike (PA) which has a supply constraint, (PA-F) is always feasible, as we now show. For each campaign $k \in \Gamma(v,i)$ we can create a pattern $p(k)$ containing exactly f_k impressions of campaign k and no other campaigns; that is, $b_{k,p(k)} = 1$ and $b_{k',p(k)} = 0$ for all $k' \neq k$. Using only such single-campaign patterns, (PA-F) has a trivial solution $y^*_{v,i,p(k)} = s_{vi} x^*_{vik}$ with dual values $\bar{\alpha}^{*(F)}_{vik} = 1,\ \forall k \in \Gamma(v,i)$. We initialize the pattern pool $\mathcal{P}_{vi}$ with only these single-campaign patterns, and from this initial solution, continue to solve (PA-F) using column generation. The corresponding pattern generating problem is the following binary knapsack problem with $|\Gamma(v,i)|$ items, which can be solved very quickly and efficiently via dynamic programming:

$$\text{(PG-F):} \qquad \psi_{vi}^{(F)} := 1 - \max\Big\{ \sum_{k\in\Gamma(v,i)} \bar{\alpha}^{*(F)}_{vik} b_k \;\Big|\; \sum_{k\in\Gamma(v,i)} f_k b_k \leq L_v,\ b_k \in \{0,1\},\ \forall k \in \Gamma(v,i) \Big\}$$

If (PG-F) concludes with $\psi^{*(F)}_{vi} < 0$, the resulting pattern improves (PA-F); we add it to $\mathcal{P}_{vi}$ and re-solve (PA-F). Otherwise, we found the optimal solution to (PA-F), and have two cases to consider.

If (PA-F) converges to optimality with $\Psi^{*(F)}_{vi} > s_{vi}$, we know the corresponding pattern assignment problem (PA) is infeasible; i.e., it is impossible to implement the solution x^*_{vik} from (RA-δ) using s_{vi} users. In this case, δ_{vi} over-estimates the attainable impression utilization, i.e., $1-\delta_{vi}$ under-estimates the fraction of impressions that must remain as excess. Consequently, we decrease δ_{vi}, re-solve (RA-δ) to produce a new reach allocation x^*_{vik}, and resume solving (PA-F) and (PG-F). To derive a good updating rule for δ_{vi}, note that the total number of impressions used (i.e., assigned to R&F ads) in pattern p is given by $\sum_k f_k b_{kp}$. Therefore, the total number of impressions used in (PA-F) at optimality is given by:

$$\sum_{p\in\mathcal{P}_{vi}} \left(\sum_{k\in\Gamma(v,i)} f_k b_{kp} \right) y^*_{vip} = \sum_{k\in\Gamma(v,i)} f_k \left(\sum_{p\in\mathcal{P}_{vi}} b_{kp} y^*_{vip} \right) =^{(6b)}= \sum_{k\in\Gamma(v,i)} f_k s_{vi} x^*_{vik}.$$

Not surprisingly, this impression count is closely tied to the solution from (RA-δ) and is known before solving (PA-F). Given that each of the $\Psi^{*(F)}_{vi}$ users provides L_v impressions, the effective impression utilization rate at the optimal solution to (PA-F) is given by $\sum_{k\in\Gamma(v,i)} f_k s_{vi} x^*_{vik} / L_v \Psi^{*(F)}_{vi}$. Based on this analysis, we suggest the following update rule:

$$\delta_{vi} \;\leftarrow\; s_{vi} X^*_{vi} / \Psi^{*(F)}_{vi} - \epsilon, \qquad (7)$$

where $X^*_{vi} = \sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x^*_{vik}$ is the left-hand side of constraint (3c) at optimality, and $\epsilon > 0$ is used to accelerate convergence.

On the other hand, if for all user types (v,i), (PA-F) converges to optimality with $\Psi^{*(F)}_{vi} \leq s_{vi}$, we have a feasible solution to all corresponding (PA) problems, and we switch to the pattern improvement phase. In this phase, for each user type (v,i), we solve (PA) and collect the optimal dual values $\bar{\alpha}^*_{vik}$ and $\bar{\beta}^*_{vi}$. Then we solve (PG) to construct a pattern with minimal reduced cost. If $\psi^*_{vi} + \bar{\beta}^*_{vi} < 0$, the resulting pattern is beneficial; we add it to $\mathcal{P}_{vi}$ (with parameters $b_{kp} = b^*_k$ and $\pi_{vip} = \pi(\mathbf{b}^*)$) and re-solve (PA). On the other hand, if $\psi^*_{vi} + \bar{\beta}^*_{vi} \geq 0$, the current solution to (PA) is optimal and we stop. Note that solving (PG) is harder than (PG-F) if $\pi(\mathbf{b})$ is not linear, however, in the pattern improvement phase we no longer solve the large-scale math program (RA-δ). Again, remember that iterations between (PA-F) and (PG-F), or (PA) and (PG), can be parallelized across user types (v,i).

Finally, at ad serving time, when a type-(v,i) user arrives for the first time, s/he is assigned pattern $p \in \mathcal{P}_{vi}$ with probability y^*_{vip}/s_{vi}. Subsequent visits of the same user are served the sequence of ads in his/her assigned pattern. If an unexpected user type (v',i') arrives, a near-optimal reach allocation $x_{v'i'k}$ is computed using Corollary 1, and a pattern is generated using the online part of Pattern-G algorithm. The full Pattern-HCG method is presented in Algorithm 4.

Remark 1: The value of δ_{vi} always decreases following update rule (7). This follows since $X^*_{vi} \leq \delta_{vi}$ due to constraint (3c), and $\Psi^{*(F)}_{vi} > s_{vi}$ whenever δ_{vi} is updated. Further, note that a decrease in impression supply at some supply node can only increase the demand burden of other supply nodes. As a result, we may need to solve (RA-δ) and update the δ_{vi} values several times before we converge.

Remark 2: A decrease in δ_{vi} implies forcing additional excess in supply node (v,i). If additional supply is not available in other supply nodes or using supply from other nodes would have a significant impact on representativeness, a δ update may cause under-delivery to increase for some campaigns. In this case, the total volume of the publisher's traffic left as excess (i.e., left for non-R&F ads) increases. However, it is also possible that after re-solving (RA-δ) with a lower δ, total under-delivery is maintained by shifting excess supply from one node to another.

Remark 3: (PA-F), which minimizes the number of users, also minimizes total excess, and thus attains the maximum impression utilization rate possible. Therefore, our update rule is conservative. See Appendix I for a proof of this behavior in the more general case of the cutting stock problem.

Remark 4: Re-solving (RA-δ) after a δ-update is quite fast, since we can warm-start SHALE using the solution from the last time we solved (RA-δ). See Theorem 2 for details.

Remark 5: We can construct bounds for the impression utilization factors δ_{vi}. Let $\delta^{\min}_{vi} = \min_{k\in\Gamma(v,i)}\{f_k\}/L_v$, which is derived from the pattern consisting of only the campaign with the smallest f_k, and let $\delta^{\max}_{vi} = \max_{b_k\in\{0,1\}}\{\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} b_k : \sum_{k\in\Gamma(v,i)} f_k b_k \leq L_v\}$, which can be computed by solving a binary knapsack problem with $|\Gamma(v,i)|$ variables. A geometric illustration of the range $[\delta^{\min}_{vi}, \delta^{\max}_{vi}]$ and how the δ_{vi} values affect the feasibility of (PA) is provided in Appendix H.

Algorithm 4 Pattern-based Hierarchical Column Generation (***Pattern-HCG***)

- **OFFLINE:**

  FEASIBILITY PHASE:

  - Initialize: $\delta_{vi} \leftarrow 1$ for all user types (v, i).
  - [1]: Solve the Reach Allocation problem (RA-δ) using ***Modified SHALE*** (Algorithm 3)
  - Parallelize: For each user type (v, i):
    - [2F]: Solve the Pattern Assignment problem (PA-F) and obtain the optimal dual values $\bar{\alpha}^{*(F)}_{vik}$.
    - [3F]: Solve the Pattern Generation problem (PG-F). If $\psi^{*(F)}_{vi} < 0$, add the generated pattern to $\mathcal{P}_{vi}$ and go to [2F]. Otherwise, continue.
    - If $\Psi^{*(F)}_{vi} > s_{vi}$, decrease δ_{vi} according to update rule (7).
  - If δ_{vi} was decreased for any user type (v, i), go to [1]. Otherwise, continue.

  PATTERN IMPROVEMENT PHASE:

  - Parallelize: For each user type (v, i):
    - [2]: Solve the Pattern Assignment problem (PA) and obtain the optimal dual values $\bar{\alpha}^{*}_{vik}$, $\bar{\beta}^{*}_{vi}$.
    - [3]: Solve the Pattern Generation problem (PG). If $\psi^{*}_{vi} + \bar{\beta}^{*}_{vi} < 0$, add the generated pattern to $\mathcal{P}_{vi}$ and go to [2]. Otherwise, stop.

- **ONLINE:** Upon a visit from user j of type (v, i):
  - If it is the first visit from user j in the planning period:
    - Set the number of arrivals $q_j \leftarrow 1$.
    - If user type (v, i) was explicitly considered as a supply node in the offline phase: Randomly draw a pattern p from the pattern pool $\mathcal{P}_{vi}$ with probability y^{*}_{vip}/s_{vi}, and denote the chosen pattern as p_j. Otherwise, construct a generalized solution x_{vik} using Corollary 1, and use the online portion of ***Pattern-G*** (Algorithm 2) to generate a corresponding pattern p_j.
  - Display the q_j'th ad in pattern p_j to user j. Set $q_j \leftarrow q_j + 1$.

Remark 6: We expect over time, a publisher may learn appropriate δ_{vi} values, and initialize with $\delta_{vi} < 1$ to speed up convergence. Nevertheless, among our numerous synthetic test cases and real industry data, we never encountered a case where it takes beyond 10 (mostly 4-6) rounds of adjustments before the reach allocation from (RA-δ) is attainable at 95% of the supply nodes.

6 Computational Experiments

Prior work in planning guaranteed targeted display advertising is impression-based; that is, it assumes publishers do not differentiate between serving 2 impressions to 1 person, or 1 impression each to 2 people. Consequently, there are no established benchmarks in the literature for comparing the performance of our methodology. In what follows, we compare Pattern-HCG with the frequency capping heuristic (FreqCap) of §3.4, which can be viewed as a reasonable proxy for how an existing impression-based ad serving system would deliver R&F campaigns, as well as with our Pattern-G heuristic from §4.1, which also serves ads using patterns but constructs patterns greedily on-the-fly rather than optimally in advance. We compare FreqCap, Pattern-G, and Pattern-HCG under different levels of sellthrough, i.e., the ratio of aggregate demand to aggregate supply (Test 1), different degrees of forecast error (Test 2), and different levels of generalized arrivals (Test 3). We also perform an out-of-sample test (Test 4) by isolating the data of a particular time period for estimation and optimization, and use other cross-sections of data for evaluating performance. We show that Pattern-HCG consistently produces 10% lower under-delivery than Pattern-G, and more than 45% lower under-delivery than FreqCap. With regard to non-representativeness, Pattern-HCG marginally outperforms Pattern-G, but both pattern-based methods outperform FreqCap by 40%.

6.1 Data[5]

Our dataset was taken from a single major vertical of *Yahoo.com* (e.g., Yahoo Mail, Yahoo News, or Yahoo Finance) and contains the following:

- The graph structure, composed of 3,844 user demographics and 925 campaigns, with 122,767 arcs (targeting specification). On average, each viewer type is targeted by 36 campaigns and each campaign targets 122 viewer types.
- The user visit history of the webpage over a period of 6 weeks. The data provides the number of page visits from each unique individual (14.7 million users), in each week, along with the exact timestamp of all visits and the demographic of each user.

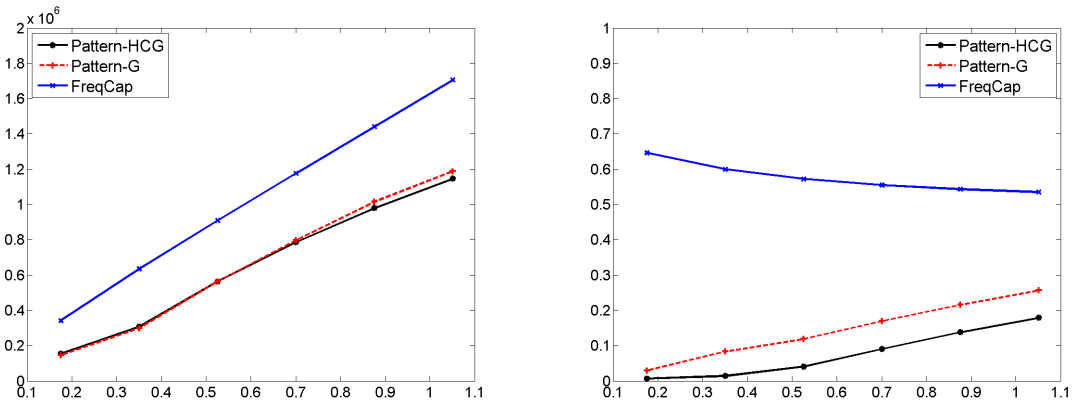
Per Yahoo's recommendation, we eliminated all users that made more than 3500 visits per week. Such users are likely to be web robots (i.e., software imitating a user) or computers shared among many individuals, and thus are not appropriate for serving R&F campaigns. This eliminated 0.1% of users and accounted for 10% of the impression traffic. We classified the remaining users into three groups $\mathcal{V} = \{$low,med,high$\}$-visiting using k-means clustering on the average number of page visits

across the 6-week period. Users with average visit count below 15 (55% of users) were considered low-, those with average visit count between 15-35 (25% of users) were considered med-, and those with average visit count above 35 (20% of users) were considered high-visiting. Then, for users of each type $v \in \{\text{low}, \text{med}, \text{high}\}$, we used the 40^{th} percentile of the page visit distribution (i.e., the threshold that is exceeded 60% of the time by users within the cluster) as the anticipated number of visits for each type-v user, and found appropriate pattern lengths of $L_v = \{10, 19, 56\}$ for the three visit types, respectively. Note that using the 40^{th} percentile for pattern lengths implies a 60% chance that each user will see all pattern slots and no ad impression planned for that user will end up as waste. Although we could chose lower percentiles to increase the probability of pattern completion, we have found lower percentiles to be overly conservative, in part due to the fact that patterns generally have some excess slots at the end anyway. We then calculated the user supply parameters s_{vi} by counting the number of users from each supply node i with visit type v that appeared in a particular week[6], and the impression supply parameters $\hat{s}_{vi}$ by counting the total number of arrivals that these s_{vi} users made. For the FreqCap algorithm, we set $\hat{s}_i = \sum_v \hat{s}_{vi}$.

Since we are only now proposing R&F campaigns, the dataset does not include relevant demand-side data. To create the demand parameters r_k, we examined existing impression-based campaign data at Yahoo and the distribution of θ_k parameters. From this distribution we randomly drew a θ_k value for each demand node. Then, in no particular order, we iterated through the demand nodes and assigned to each node k a θ_k-fraction of the remaining supply from each node $(v, i) \in \Gamma(k)$ to produce an initial estimate for r_k; such a construction parallels the so-called *High Water Mark* algorithm discussed in Bharadwaj et al. (2012). Finally, we scaled and rounded the r_k values to yield a sellthrough of approximately 88%. We generated frequency targets f_k independently at random between 1 and 25, following a positively-skewed bell-shaped distribution that peaked around a frequency of 6. In all tests, we used penalty weights $w_k = 1$ and $c_k = 3$ for all campaigns, as per Yahoo's suggestion, avoiding under-delivery (which has a direct revenue consequence) is more important than maximizing representativeness.

6.2 Results

All algorithms were implemented in Matlab® and run in a parallelized environment with 32 cores at 2.3GHz each. The runtimes observed under Pattern-HCG are as follows. Each round of solving the reach allocation problem (RA-δ) using Modified SHALE took 30-60 seconds, and each round of pattern generation and assignment took about 25 minutes (about 4 seconds per supply node (v, i), though 54% of nodes completed their CG within 1 second). Typically, it took only 4-6 iterations of the feasibility phase to produce patterns that attained the reach assignment from (RA-δ) at 95% of the supply nodes. Therefore, on average, each run of Pattern-HCG took about two hours. In the final solution, we observe close to 130,000 unique patterns, ranging from 1 to 121 with an average of 12 patterns for each user type (v, i). In contrast, the offline phase of Pattern-G solves (RA-δ) only once, and consequently takes only 30-60 seconds. More details about each test and the results appears below.



(a) Non-representativeness Penalty

(b) Under-delivery Fraction $\sum_k u_k / \sum_k r_k$

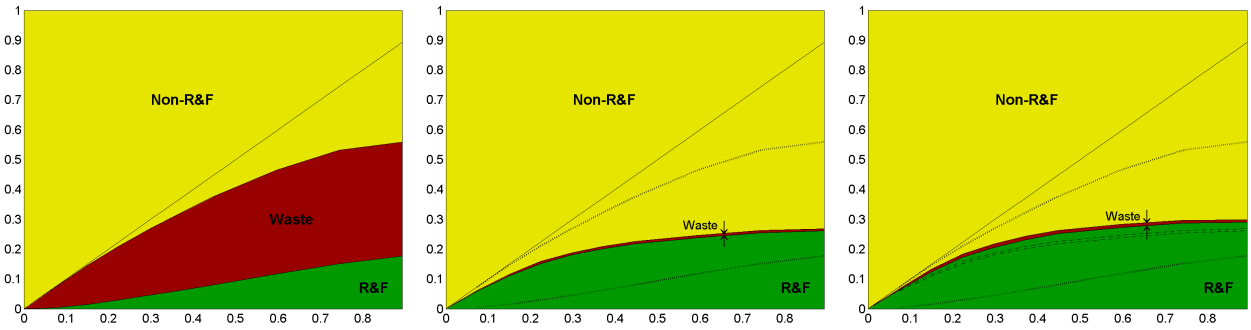
Figure 4: Performance of our three methods at different levels of sellthrough $\mathcal{S}_{R\&F}$.

Test 1: Performance at Different Sellthrough Levels

Sellthrough, defined as the ratio of aggregate demand to aggregate supply, is a well-known performance metric in marketing and retail operations. It measures supply scarcity, and how hard it is to satisfy demand. We consider two sellthrough measures, $\mathcal{S}_{Tot} = \sum_k f_k r_k / \sum_{v,i} \hat{s}_{vi}$, which is measured in terms of the total impression traffic, and $\mathcal{S}_{R\&F} = \sum_k f_k r_k / \sum_{v,i} L_v s_{vi}$ which is measured in terms of the proportion of impression traffic that is eligible for R&F campaigns. In our dataset, $\sum_{v,i} L_v s_{vi} / \sum_{v,i} \hat{s}_{vi} \simeq 0.43$; therefore, the two measures are related via $\mathcal{S}_{Tot} = 0.43 \mathcal{S}_{R\&F}$. To vary sellthrough, we scale all r_k values by a constant factor. In this section, we assume perfect supply forecasts to isolate the effect of a change in sellthrough.

Figure 4 compares the non-representativeness and under-delivery we observed for each method at different levels of sellthrough $\mathcal{S}_{R\&F}$. As expected, performance generally declines as sellthrough increases and the instance becomes more constrained. Note that with ample supply (very low sellthrough), Pattern-HCG (solid black line) has only marginally better under-delivery than Pattern-G (dashed red line); however, the performance gap widens at higher sellthrough levels. Indeed, for $\mathcal{S}_{R\&F} \geq 0.4$ Pattern-HCG produces 10% less under-delivery than Pattern-G, which at $\mathcal{S}_{R\&F} = 0.7$ constitutes a reduction in under-delivery by nearly half and at $\mathcal{S}_{R\&F} = 0.88$ constitutes a reduction of nearly one-third. Beyond a certain sellthrough level (about 55%), additional reach cannot be packed into the limited pattern space, and therefore, under-delivery of both pattern-based methods increase linearly, with a mild slope. In contrast, the performance of FreqCap is clearly inferior to both Pattern-G and Pattern-HCG, but somewhat paradoxically its under-delivery improves as sellthrough increases. This is due to the fact that higher sellthrough requires a higher proportion of supply to be allocated, thereby increasing the probabilities x_{ik} that any campaign k is drawn upon a user visit. This increases the probability that all f_k impressions of campaign k are successfully delivered to the user.

Figure 5 demonstrates the proportion of impressions, out of the full supply $\sum_{v,i} \hat{s}_{vi}$ served to



(a) FreqCap (b) Pattern-G (c) Pattern-HCG

Figure 5: At different levels of sellthrough $\mathcal{S}_{Tot}$ (horizontal axis), we show the proportion of impressions assigned to Non-R&F campaigns (yellow), assigned to R&F campaigns that were wasted (red), and assigned to R&F campaigns that were billable (green). Dotted lines show the boundaries of regions in the subfigures to the left, allowing easy comparisons left-to-right.

R&F campaigns (lower region), impressions wasted due to R&F campaigns not reaching their target frequency (middle region), and impressions left as excess for non-R&F planning (top region), at each sellthrough level. We use $\mathcal{S}_{Tot}$ for measuring sellthrough here as it makes the plots easier to interpret: The total impression demand increases along the 45-degree line, starting from the origin. The union of the two green and red areas shows the fraction of R&F impression demand, $\sum_k f_k r_k$, allocated by (IA), (RA), and (RA-δ), respectively in subfigures (a), (b), and (c). The deviation below the 45-degree line can be interpreted as planned under-delivery, $\sum_k f_k u_k$, measured in impressions.

Figure 5(a) shows that FreqCap allocates the most number of impressions to R&F campaigns, but nearly 2/3 of these impressions fall short of the target frequency at the user level, and therefore end up as waste. Figure 5(b) shows that Pattern-G, which uses a pattern-based allocation mechanism, does a much better job of reducing waste than FreqCap, with waste below 3% at all levels of sellthrough. Finally, Figure 5(c) shows that Pattern-HCG is able to keep waste low while additionally increasing the proportion of impressions successfully served to R&F campaigns (the green region is larger). Although from this figure it does not seem like there is a large difference between how Pattern-HCG and Pattern-G deliver R&F impressions, the difference is enough for Pattern-HCG to achieve substantially less reach under-delivery than Pattern-G (recall Figure 4). Note that $\mathcal{S}_{Tot} = 0.43$ implies $\mathcal{S}_{R\&F} = 1$ which is the absolute maximum sellthrough we can hope to serve without under-delivery, and this is only possible when: 1) campaigns' targeting criteria does not restrict our ability to allocate the entire user traffic, 2) the frequency requirements can be perfectly packed into pattern lengths without excess, and 3) the problem is relaxed to allow each user to be reached multiple times by the same campaign. Consequently, a sellthrough of $\mathcal{S}_{Tot} \simeq 0.3$ should be considered quite high, practically speaking.

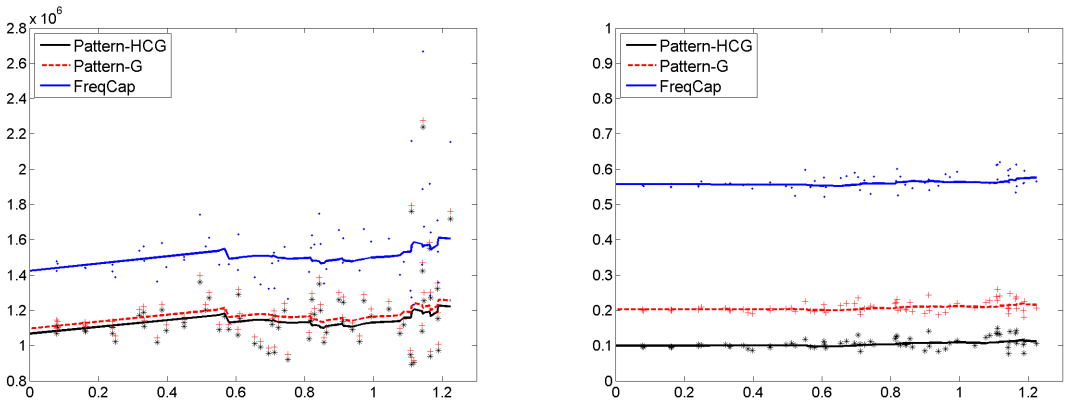
Test 2: Robustness to Forecast Errors

Our offline optimization methodology produces a serving plan according to the forecasted supply of users, s_{vi}. The actual number of users that visit the publisher's website, denoted $s_{vi}^{(a)}$, is uncertain and may differ from the forecast. Therefore, it is important to check the robustness of our solutions to forecast error. In this test, we use the actual observed traffic s_{vi} and $\hat{s}_i$ to produce a plan under our three algorithms. Then, we evaluate the performance of these solutions under random arrival streams that are created in the following way. First, we add Gaussian noise to every supply node's forecast, i.e., $s_{vi}^{(a)} \leftarrow (1 + c \cdot \varepsilon_{vi}) s_{vi}$ where ε_{vi} is a standard normal random variable (with a mean of zero and a standard deviation of one), and c is the desired coefficient of variation (CV) of the Gaussian noise, which we take to be identical for all supply nodes. We vary c to produce arrival streams that have different degrees of forecast error. Negative supply values, if produced, are truncated to zero and then we normalize the arrival stream to keep the aggregate level of traffic invariant. This way, we isolate effect of variability in the sizes of supply nodes from changes in sellthrough, which we tested separately in Test 1. Finally, we probabilistically round each generated user count $s_{vi}^{(a)}$ to a neighboring integer (e.g., 5.3 is rounded to 5 with probability 0.7, and to 6 with probability 0.3), to yield integer $s_{vi}^{(a)}$ values while keeping the aggregate supply stable. We generate the number of visits for each user using the empirical probability distributions $\phi_v(\cdot)$ obtained from the dataset after clustering user visit types. This is our only computational test in which we do not use the observed arrival stream from the data to evaluate the performance of our solution. Note that following the truncation and normalization steps, the CV parameter c is no longer a reliable measure of forecast noise. Instead, we use Mean Absolute Percentage Error (MAPE) to measure how the random arrival stream $s_{vi}^{(a)}$ differs from the forecast s_{vi}:

$$\text{MAPE} \;=\; \frac{1}{|\mathcal{I}||\mathcal{V}|} \sum_{(v,i)} \frac{|s_{vi} - s_{vi}^{(a)}|}{s_{vi}}.$$

Figure 6 shows the performance of each method in terms of non-representativeness and under-delivery, under different degrees of forecast error. Forecast MAPE, along the horizontal axis, ranges from 0 to about 1.3. Note that a MAPE of 1 indicates that on average, the actual number of users observed in each supply node differed by 100% from its forecast. Each dot corresponds to a different random instance of the arrival stream[7]. The curves are basic moving averages which illustrate the overall trend.

At our baseline sellthrough of 88%, we find that the average under-delivery of Pattern-HCG (solid black line) is consistently half that of Pattern-G (dashed red line), and one-fifth that of FreqCap (solid blue line), and that this relationship roughly holds at all all degrees of forecast MAPE. The non-representativeness penalty obtained by Pattern-G is comparable to that of Pattern-HCG; both outperform FreqCap by a consistent 30% at all levels of forecast noise. Our experiments show that the under-delivery and non-representativeness performance of all algorithms is quite robust to forecast error.



(a) Non-representativeness Penalty

(b) Under-delivery Fraction $\sum_k u_k / \sum_k r_k$

Figure 6: Performance under noisy forecasts, as a function of mean absolute percentage error (MAPE). Each dot corresponds to a different random arrival stream.

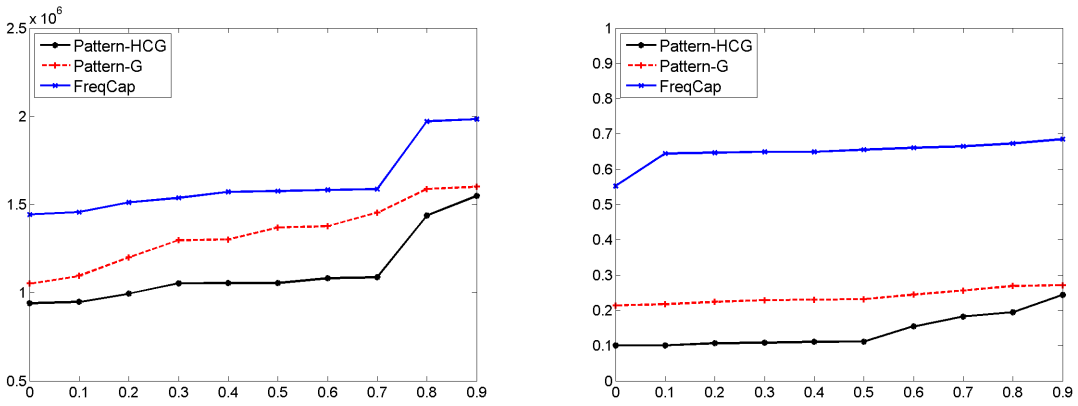
Test 3: Robustness to Graph Sampling Error (Generalizability)

As described in §3, generalizability is important when there are a large number of demographics, and only the most important subset of demographics (e.g., those with enough historical data to accurately forecast) are used to produce the optimal ad allocation. If an arriving user belongs to a demographic that was not explicitly used to construct the optimal ad allocation, we use Corollary 1 to produce a near-optimal solution and serve ads accordingly. Figure 7 plots the under-delivery and non-representativeness performance of FreqCap, Pattern-G, and Pattern-HCG under different levels of generalized arrivals. The horizontal axis shows the proportion of supply nodes we omitted uniformly at random from the original graph when solving our offline plans. In each case, we scale the supply of remaining nodes up to keep the sellthrough level constant at 88% which allows us to isolate the effect of generalizability. We then test the performance of the obtained solution on the full arrival stream observed in the data (i.e., there is no forecast error, $s_{vi}^{(a)} = s_{vi}$).

With regard to under-delivery, Pattern-HCG (solid black line) outperforms Pattern-G (dashed red line) by 10% when no generalized arrivals occur. This performance gap decreases as the proportion of generalized arrivals increases, and is minimal once the proportion of generalized arrivals reaches 90%, i.e., only 10% of the full graph is represented by the sample used at planning/optimization time. Again, FreqCap exhibits subpar performance with 5 times higher under-delivery than Pattern-HCG. With regard to non-representativeness, Pattern-HCG outperforms Pattern-G by 10-30% and FreqCap by 30-50%.

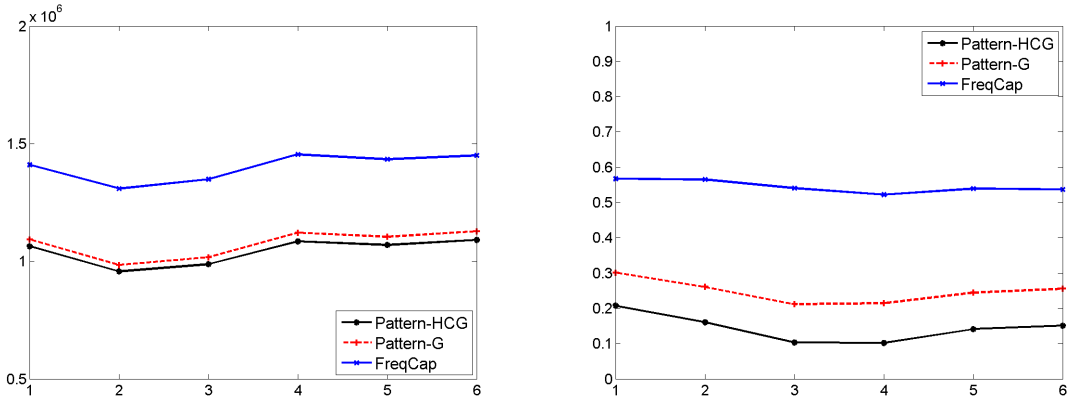
Test 4: Out of Sample Testing

In practice, there are several sources of uncertainty at the planning stage. These include the number of users s_{vi} of each type (v, i), the number of visits that each individual user makes, as well as the aggregate volume of users and impressions across all user types, which affects sellthrough. For this



(a) Non-representativeness Penalty (b) Under-delivery Fraction $\sum_k u_k / \sum_k r_k$

Figure 7: Performance under generalized arrivals. The horizontal axis shows the fraction of supply nodes omitted from the graph at optimization/planning time.



(a) Non-representativeness Penalty (b) Under-delivery Fraction $\sum_k u_k / \sum_k r_k$

Figure 8: Performance measured in weeks 1-6, using data from week 4 for parameter estimation and optimization.

test, we split our dataset by weeks, numbered 1 through 6. We then use the data from week 4 to estimate parameters and obtain the optimal solutions using FreqCap, Pattern-G, and Pattern-HCG. Then we apply the week-4 solutions to the arrival streams from each of the other weeks {1,2,3,5,6}. This provides us with 5 out-of-sample instances of $s_{vi}^{(a)}$ along with a number of visits per each user to test our solutions from week 4. Results are shown in Figure 8. This test can be thought of as a robustness check to confirm the viability of our approach in practice. It assumes that the most naïve forecasting system is employed by the publisher, i.e., one that uses a historical observation from another period as its forecast. We observe that the relative performance gaps are consistent among the three methods across all 6 weeks, with Pattern-HCG consistently performing best.

7 Conclusions

In line with recent industry trends and growing attention to reach, personalized marketing, and storyboarding, we introduced and modeled, for the the first time, *guaranteed reach and frequency contracts* for online targeted display advertising, and proposed a novel mechanism for ad planning and delivery that employs pre-generated *patterns* to schedule the exact sequence of ads for each individual user. We showed that our model can be implemented efficiently using a two-phase algorithm that employs column generation in a hierarchical scheme with three parallelizable components. Our optimization framework strives for aggregate quality of ad delivery (i.e., retained revenue and uniform spread of campaigns among their target audience) as a primary objective, as well as disaggregate quality (e.g., diversity and pacing of ads over time as delivered to each individual) as a secondary objective. Exponential growth of mobile device usage and new identifier technologies that allow publishers to accurately track individuals over time contribute to making our modeling approach relevant and practical.

Based on our computational testing on real industry data, we conclude that our use of column generation for constructing patterns together with our mechanism for tuning impression utilization factors results in significantly better performance (10% and 45% less under-delivery and better representativeness compared to our pattern-based greedy heuristic and frequency capping, respectively). In practice, if time is limited, one may employ the feasibility phase of our Pattern-HCG method and make a limited number of δ-adjustments, and then jump to a reasonable solution using Pattern-G. Nevertheless, we expect that the runtime of Pattern-HCG is within practical applicability for offline planning in the industry, assuming proper parallelization and specialized coding for large instances. Even though our main model is deterministic, our computational tests show that our solution is indeed robust to forecast error and randomness in user arrivals. Our probabilistic model, presented in Appendix D, that explicitly models the randomness of user arrivals in the pattern generation process, can create more robust solutions with longer, less conservative, pattern lengths at the expense of additional computation times.

There are a number of open questions that are left for future research. First, it would be interesting to derive a competitive ratio (i.e., worst-case optimality gap) under some relevant input model. However, doing so would also require developing a new input model that is structurally useful (i.e., leads to elegant theoretical proofs) which is tailored for the R&F ad planning problem. The existing input models (e.g., for a survey see Mehta 2012) are all impression-based and do not specify how to characterize the uncertainty of the arrival stream for specific users. Moreover, if sufficiently suitable characterizations of the arrival process can be identified which have parameters which are relatively easy to estimate from data, it may also be of interest to develop a Stochastic Dynamic Programming (DP) model for the R&F ad planning problem. There are a number of issues that would need to be overcome when modeling this problem as a Stochastic DP. First, such a model would have an exponentially large state space, since an optimal dynamic policy would need to know, at each impression arrival, the number of times each user j has seen each ad campaign k up to the current point in time. Second, our problem has both primary and secondary objectives;

this would be challenging to handle with DP. Third, many of our objective functions such as representativeness (aggregate quality) and user-level pacing (disaggregate quality) are non-linear multivariate functions that measure how ads are delivered over the full planning period. They are non-separable in individual impression assignments; i.e., they do not easily decompose into increments of reward or cost that accumulate after each impression assignment. Sniedovich (2010) offers some suggestions to tackle non-separable objectives, but the methods he describes either involve state-augmentation (which would grow the already exponential state space) or iteratively solving the DP numerous times (which increases solution times). It would be interesting to develop some clever modeling approaches to skirt these and other challenges**.**

Finally, we note that our pattern-based approach for serving web advertisements which allows for personalized planning can also be applied to other forms of technology-enabled advertising, including digital TV, online videos, and in-game advertising. Moreover, our Pattern-HCG method can be adapted to other planning and allocation problems with primary (aggregate) and secondary (disaggregate) goals, where optimizing with respect to the primary objective is structurally similar to a transportation problem (bipartite allocation).

Endnotes

1. Facebook, Google, and Yahoo had net U.S. display ad revenues of $5.29, $3.03, and $1.23 billion in 2014 (eMarketer 2015).

2. Denoting $q_k = E_{\boldsymbol{\xi}}[\sum_j z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))] = \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}$ and $\mu_k = q_k/S_k$, we have $Var(\sum_{j\in J} z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))) = \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}(1-x_{vik}) \leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\mu_k(1-\mu_k) = q_k(1-\mu_k)$, where the first equality follows from the fact that $\sum_{j\in J(v,i)} z_{jk}(\tilde{\pi}(\boldsymbol{\xi})) \sim Binomial(s_{vi}, x_{vik})$, and the inequality is due to Jensen's Inequality. The Coefficient of Variation (CV) of the number of users that will be reached, $\sum_{j\in J} z_{jk}(\tilde{\pi}(\boldsymbol{\xi}))$, is therefore $\sqrt{q_k(1-\mu_k)}/q_k = \sqrt{1/q_k - 1/S_k}$. This CV is always less than or equal to 1, and in most cases, much, much smaller. For example, with only $q_k = 100$ users planned to be allocated to campaign k, the CV is already below 0.1; with $q_k = 1$ million users, the CV is below 0.001.

3. We also need an estimate for $\delta_{v'i'}$ to compute $\beta_{v'i'}$. Any value within the bounds defined in Remark 5 of Section 5.4 would be reasonable. Our numerical experiments show that picking $\delta_{v'i'} = \delta^{\max}_{v'i'}$ and then applying Pattern-G produces a good solution.

4. This runtime corresponds to the problem instances we study in §6, which have $|\Gamma(v,i)|$ between 1 and 442 with an average connectivity of 36 campaigns per viewer type, and three pattern lengths $L_v \in \{10, 19, 56\}$.

5. We intend to publish our dataset, e.g., through Yahoo Labs Webscope, so it is accessible to the operations research community for future developments and benchmarking. The process of clearing the release of the data is still ongoing at the time of this submission. The reported dataset and results are subject to anonymization and deliberately incomplete to not reflect the real portfolio of Yahoo at any particular time.

6. We use week 4 as it gave us a slightly higher number of supply nodes with $s_{vi} > 0$, i.e., a more complete graph, compared to other weeks.

7. The assignment of patterns to users is a random process (pattern p is chosen for a user of type (v,i) with probability y^*_{vip}/s_{vi}) and differs in each run of the simulation, which has a slight impact on the performance. For each arrival stream, the solution was simulated multiple times to accurately report the performance.

References

Abrams, Z., S. S. Keerthi, O. Mendelevitch, and J. A. Tomlin (2008). Ad delivery with budgeted advertisers: A comprehensive LP approach. *Journal of Electronic Commerce Research 9*(1), 16–32.

Adaptly (2014, May). A research study on sequenced for call to action vs. sustained call to action. Available online at: `http://adaptly.com/wp-content/uploads/2014/11/Adaptly-Refinery29-White-Paper-2014.pdf`.

Agrawal, S., Z. Wang, and Y. Ye (2014). A dynamic near-optimal algorithm for online linear programming. *Operations Research 62*(4), 876–890.

Ahuja, R. K., T. L. Magnanti, and J. B. Orlin (1993). *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Araman, V. F. and K. Fridgeirsdottir (2010). A uniform allocation mechanism and cost-per-impression pricing for online advertising. *Working paper*.

Balseiro, S. R., J. Feldman, V. Mirrokni, and S. Muthukrishnan (2014). Yield optimization of display advertising with ad exchange. *Management Science 60*(12), 2886–2907.

Besbes, O. and C. Maglaras (2012). Dynamic pricing with financial milestones: feedback-form policies. *Management Science 58*(9), 1715–1731.

Bharadwaj, V., P. Chen, W. Ma, C. Nagarajan, J. Tomlin, S. Vassilvitskii, E. Vee, and J. Yang (2012). SHALE: An efficient algorithm for allocation of guaranteed display advertising. In *Proceedings of the 18th ACM international conference on knowledge discovery and data mining*, pp. 1195–1203.

Bollapragada, S., M. R. Bussieck, and S. Mallik (2004). Scheduling commercial videotapes in broadcast television. *Operations Research 52*(5), 679–689.

Brusco, M. (2008). Scheduling advertising slots for television. *Journal of the Operational Research Society 59*(10), 1363–1372.

Buchbinder, N., M. Feldman, A. Ghosh, and J. S. Naor (2011). Frequency capping in online advertising. In *Algorithms and Data Structures*, pp. 147–158. Springer.

Campbell, M. C. and K. L. Keller (2003). Brand familiarity and advertising repetition effects. *journal of Consumer Research 30*(2), 292–304.

Chandler-Pepelnjak, J. and Y.-B. Song (2003). Optimal frequency – the impact of frequency on conversion rates. Atlas Digital Insights. Available online at: `http://advertising.microsoft.com/wwdocs/user/en-us/researchlibrary/researchreport/OptFrequency.pdf`.

Chen, P., W. Ma, S. Mandalapu, C. Nagarjan, J. Shanmugasundaram, S. Vassilvitskii, E. Vee, M. Yu, and J. Zien (2012). Ad serving using a compact allocation plan. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 319–336.

Chickering, D. M. and D. Heckerman (2003). Targeted advertising on the web with inventory management. *Interfaces 33*(5), 71–77.

Desaulniers, G., J. Desrosiers, and M. M. Solomon (2005). *Column generation*, Volume 5. New York: Springer.

Devanur, N. R. and T. P. Hayes (2009). The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 71–78. ACM.

Devanur, N. R., K. Jain, B. Sivan, and C. A. Wilkens (2011). Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, pp. 29–38. ACM.

eMarketer (2009, July). The great GRP debate. Available online at: `http://www.emarketer.com/Articles/Print.aspx?R=1007174`.

eMarketer (2014, June). How do you combine TV and digital video? Available online at: `http://www.emarketer.com/Articles/Print.aspx?R=1010900`.

eMarketer (2015, March). Facebook and twitter will take 33% share of us digital display market by 2017. Available online at: `http://www.emarketer.com/Articles/Print.aspx?R=1012274`.

Feichtinger, G., R. F. Hartl, and S. P. Sethi (1994). Dynamic optimal control models in advertising: recent developments. *Management Science 40*(2), 195–226.

Feldman, J., M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein (2010). Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*, pp. 182–194. Springer.

Feldman, J., A. Mehta, V. Mirrokni, and S. Muthukrishnan (2009). Online stochastic matching: Beating 1-1/e. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pp. 117–126. IEEE.

Ghosh, A., P. McAfee, K. Papineni, and S. Vassilvitskii (2009). Bidding for representative allocations for display advertising. In *Workshop on Internet and Network Economics (WINE)*, pp. 208–219. LNCS 5929, Berlin: Springer.

Gilmore, P. C. and R. E. Gomory (1961). A linear programming approach to the cutting-stock problem. *Operations Research 9*(6), 849–859.

Goel, G. and A. Mehta (2008). Online budgeted matching in random input models with applications to adwords. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 982–991. Society for Industrial and Applied Mathematics.

Hojjat, A., J. Turner, S. Cetintas, and J. Yang (2014). Delivering guaranteed display ads under reach and frequency requirements. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 2278–2284.

Interactive Advertising Bureau (2015, April). IAB 2014 full-year internet advertising revenue report. Available online at: `http://www.iab.net/research/industry_data_and_landscape/adrevenuereport`.

Jones, D. and M. Tamiz (2010). *Practical goal programming*, Volume 141. Springer.

Kattula, J., J. Lewis, and J. Dailey (2015). Behind the buzz: People-based marketing defined. Atlas Solutions, LLC. Available online at: `https://atlassolutionstwo.files.wordpress.com/2015/05/atlas_white_paper_people-based_marketing_may_2015.pdf`.

Kubiak, W. and S. Sethi (1991). A note on "level schedules for mixed-model assembly lines in just-in-time production systems". *Management Science 37*(1), 121–122.

Kubiak, W. and S. P. Sethi (1994). Optimal just-in-time schedules for flexible transfer lines. *International Journal of Flexible Manufacturing Systems 6*(2), 137–154.

Langheinrich, M., A. Nakamura, N. Abe, T. Kamba, and Y. Koseki (1999). Unintrusive customization techniques for web advertising. *Computer Networks 31*(11), 1259–1272.

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

Manshadi, V. H., S. O. Gharan, and A. Saberi (2012). Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research 37*(4), 559–573.

Martello, S. and P. Toth (1990). *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.

Mehta, A. (2012). Online matching and ad allocation. *Theoretical Computer Science 8*(4), 265–368.

Mehta, A., A. Saberi, U. Vazirani, and V. Vazirani (2007). Adwords and generalized online matching. *Journal of the ACM (JACM) 54*(5), 22.

Mookerjee, R., S. Kumar, and V. S. Mookerjee (2012). To show or not show: Using user profiling to manage internet advertisement campaigns at chitika. *Interfaces 42*(5), 449–464.

Najafi Asadolahi, S. and K. Fridgeirsdottir (2014). Cost-per-click pricing for display advertising. *Manufacturing & Service Operations Management, Forthcoming*.

Nakamura, A. and N. Abe (2005). Improvements to the linear programming based scheduling of web advertisements. *Electronic Commerce Research 5*(1), 75–98.

Roels, G. and K. Fridgeirsdottir (2009). Dynamic revenue management for online display advertising. *Journal of Revenue & Pricing Management 8*(5), 452–466.

Salomatin, K., T.-Y. Liu, and Y. Yang (2012). A unified optimization framework for auction and guaranteed delivery in online advertising. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 2005–2009.

Sethi, S. P. (1977). Dynamic optimal control models in advertising: a survey. *SIAM review 19*(4), 685–725.

Sniedovich, M. (2010). *Dynamic programming: foundations and principles*. CRC press.

Tomlin, J. A. (2000). An entropy approach to unintrusive targeted advertising on the web. *Computer Networks 33*(1), 767–774.

Turner, J. (2012). The planning of guaranteed targeted display advertising. *Operations Research 60*(1), 18–33.

Vee, E., S. Vassilvitskii, and J. Shanmugasundaram (2010). Optimal online assignment with forecasts. In *Proceedings of the 11th ACM conference on Electronic commerce*, pp. 109–118.

Warc (2015, August). Marketers rely on 'broken' cookies. Available online at: `http://www.warc.com/LatestNews/News/EmailNews.news?ID=35181`.

Yang, J., E. Vee, S. Vassilvitskii, J. Tomlin, J. Shanmugasundaram, T. Anastasakos, and O. Kennedy (2010). Inventory allocation for online graphical display advertising. *arXiv preprint arXiv:1008.3551*.

A Table of Notation

Sets and Indices	
$k \in \mathcal{K}$	Advertising campaigns.
$i \in \mathcal{I}$	User demographics, based on targeting attributes.
$v \in \mathcal{V}$	User visit-types, based on the minimal number of visits expected from the user (see: L_v).
$p \in \mathcal{P}_{vi}$	Patterns created for users of visit-type v and demographic i.
ℓ	$\in \{1, ..., L_v\}$ Slots in the pattern (resp., number of visits made by a user of visit-type v).
$\mathcal{T}$	Targeting: $(i,k) \in \mathcal{T}$ implies user demographic i meets the targeting criteria of campaign k.
$\hat{\Gamma}(i)$	$= \{k \mid (i,k) \in \mathcal{T}\}$ Set of campaigns that target user demographic i.
$\hat{\Gamma}(k)$	$= \{i \mid (i,k) \in \mathcal{T}\}$ Set of user demographics that meet the targeting criteria of campaign k.
$\Gamma(v,i)$	$= \{k \mid (i,k) \in \mathcal{T},\ f_k \leq L_v\}$ Set of campaigns eligible for type-(v,i) user, i.e., demographic i is targeted and the frequency f_k is within the number of visits, L_v, anticipated from this user.
$\Gamma(k)$	$= \{(v,i) \mid (i,k) \in \mathcal{T},\ L_v \geq f_k\}$ Set of user types (v,i) targeted by campaign k and anticipated (with high probability) to make more visits than the frequency requirement f_k.

Parameters	
$\hat{d}_k$	Demand: Number of impressions desired by campaign k (impression-based contract).
r_k	Reach: Number of unique users desired to be reached by campaign k (R&F contract).
f_k	Frequency: Number of times a user must see campaign k's ad to be counted as reached.
$c_k(\hat{c}_k)$	Cost per unit of under-delivery for campaign k measured in users (impressions).
$w_k(\hat{w}_k)$	Penalty weight for non-representativeness of campaign k measured in users (impressions).
$\hat{s}_i$	Supply of impressions from users of demographic i.
s_{vi}	Supply of unique users of demographic i with visit-type v.
$\hat{S}_k$	$= \sum_{i \in \Gamma(k)} \hat{s}_i$ Total impression traffic eligible for campaign k.
S_k	$= \sum_{(v,i) \in \Gamma(k)} s_{vi}$ Total user traffic eligible for campaign k.
$\hat{\theta}_k$	$= \hat{d}_k / \hat{S}_k$ Ideal representative fraction of impressions $i \in \Gamma(k)$ for campaign k.
θ_k	$= r_k / S_k$ Ideal representative fraction of users $(v,i) \in \Gamma(k)$ for campaign k.
$\phi_v^{(\ell)}$	Probability that a type-v user will make exactly $\ell \in \{1, ..., \bar{L}_v\}$ visits.
$\Phi_v(\ell)$	$= \sum_{\ell'=0}^{\ell} \phi_v^{(\ell')}$ is the CDF of $\phi_v^{(\ell)}$.
L_v	$= \Phi_v^{-1}(\varepsilon)$ (integer): Appropriate pattern length for a user with visit-type v. Any user of visit-type v visits at least L_v times and sees the entire pattern with a high probability $1-\varepsilon$. We also refer to L_v as the *anticipated number of visits* from a user with visit-type v.
b_{kp}	(binary): 1 if f_k impressions of campaign k are included in pattern p, and 0 otherwise. We use $\mathbf{b}$ to denote the entire decision vector $\big(b_k\big)_{k \in \Gamma(v,i)}$ in a sub-problem (v,i).
π_{vip}	Unit cost of using pattern $p \in \mathcal{P}_{vi}$ (captures poor pacing, lack of diversity, and/or excess). This is measured using a function $\pi(\mathbf{b})$ described in Appendix B.
δ_{vi}	Proportion of type-(v,i) impressions usable when serving with patterns (after trim loss). $\delta_{vi}^{\min}$ and $\delta_{vi}^{\max}$ give a priori lower- and upper-bounds on the value of δ_{vi}. The values of the δ_{vi} parameters are tuned within our algorithm.

Decision Variables	
Impression Allocation (IA)	
$\hat{x}_{ik}$	Proportion of impressions of demographic i allocated to campaign k.
$\hat{u}_k$	Under-delivery of campaign k (number of impressions assigned to k short of its demand $\hat{d}_k$).
Reach Allocation (RA)	
x_{vik}	Proportion of users of type (v,i) to be reached by campaign k.
u_k	Under-delivery of campaign k (number of unique users assigned to k short of its reach target r_k).
Pattern Assignment (PA)	
y_{vip}	Number of users of type (v,i) served using pattern $p \in \mathcal{P}_{vi}$.
Pattern Generation (PG)	
b_k	(binary): 1 if we include (f_k impressions of) campaign k in this pattern, and 0 otherwise. Becomes the parameter b_{kp} once the generated pattern is stored (with index p).

B Pattern Quality Metrics

In this section we elaborate on possible choices for the cost measure $\pi(\mathbf{b})$ and their impact on the complexity of solving the pattern generation problem (PG). For example, we can define $\pi(\mathbf{b})$ to produce patterns that: 1) are diverse, to expose the user to a large variety of ads; 2) have some amount of excess, making the plan robust to uncertainty in the number of visits from each user, or 3) are well-paced, that is, if campaign k is included in the pattern, then its f_k impressions should be uniformly spread across the pattern's L_v slots. Additionally, we show how to ensure campaigns from competing brands do not appear in the same pattern.

1. Maximizing diversity

Diversity is measured as the number of campaigns in the pattern. The following linear cost measure penalizes lack of diversity:

$$\pi_{diversity}(\mathbf{b}) \;=\; -\sum_{k\in\Gamma(v,i)} b_k$$

As discussed in §5.3, (PG) is efficiently solvable when $\pi(\mathbf{b})$ is linear.

2. Maximizing or minimizing excess

The following linear cost measure penalizes the slack of capacity constraint (5b), and thus the amount of excess in the pattern:

$$\pi_{excess}(\mathbf{b}) \;=\; \left(L_v - \sum_{k\in\Gamma(v,i)} f_k b_k\right) \bar{c}_{vi}$$

The parameter $\bar{c}_{vi}$ captures the opportunity cost of replacing a more expensive guaranteed R&F ad with a non-guaranteed ad for a user of type (v,i).

During Pattern-HCG's pattern improvement phase, the total amount of excess at each supply node (v,i) stays fixed at $L_v s_{vi} - \sum_{k\in\Gamma(v,i)} f_k s_{vi} x^*_{vik}$ which is determined by the reach allocation problem (RA-δ). However, optimizing the number of excess slots within patterns affects both the number of unique patterns in each supply pool $\mathcal{P}_{vi}$, as well as the number of times each pattern is used. Specifically:

- *Maximizing excess* creates patterns that are less likely to waste impressions. Excess provides a buffer that makes the pattern robust to uncertainty in the number of visits made by each user. As well, although in expectation non-guaranteed ads have lower value than R&F, it could happen that due to a particular user's recent browsing behavior (e.g., shopping for a particular item), this user's impressions become very valuable in the non-guaranteed marketplace. To hedge against such opportunities, the publisher may wish to reserve excess impressions for each user.

- *Minimizing excess* creates patterns that are better-packed with R&F campaigns. As a result, we tend to use fewer patterns, i.e., pattern pools are smaller, reducing the memory load on the ad

server. As well, we need fewer unique users to deliver the reach allocation x^*_{vik}, making the plan more robust to uncertainty in the supply of unique users, s_{vi}.

So there are pros and cons to having excess and the choice of maximizing or minimizing excess should depend on the solution structure desired by the publisher, and the stability of user traffic and number of visits per user. We expect this to vary from one publisher to another. In both cases, π_{excess} is a linear function of the decision variables b_k and thus (PG) is efficiently solvable. That said, we expect that a probabilistic model, such as the one we propose in Appendix D which explicitly takes into account the randomness of user arrivals when generating patterns, would eliminate the need for considering either minimization or maximization of excess as a pattern quality metric.

3. User-level pacing of ads

The existing research that explicitly considers smooth/uniform delivery of campaigns focuses on the cumulative impressions received by each campaign in aggregate (Araman and Fridgeirsdottir 2010), budget depletion, or financial milestones (Besbes and Maglaras 2012) and is not at the individual user level. We now discuss several approaches for measuring and optimizing the extent to which impressions of a campaign are well-spread at individual user level. This is accomplished by measuring and optimizing the spread of a campaign over the slots of a pattern. The function $\pi_{pacing}(\mathbf{b})$ which penalizes deviations from a uniform spread, by itself involves solving an inner optimization problem to sequence the f_k impressions of the campaigns in the pattern (i.e., campaigns with $b_k = 1$). This inner optimization problem has been studied in two streams of papers which we now review. These two approaches differ based on how they define uniformity and how they measure and penalize non-uniformity of the arrangement, which leads to differences in solution structure and computational complexity. For convenience, we use our notation to describe their models.

Kubiak and Sethi (1991) consider the optimal scheduling of a multi-product assembly line in which each product k has a fixed known demand f_k and is expected to be produced at a constant rate f_k/L_v throughout the production horizon L_v. Within the context of our problem, let $z_{k\ell} \in \{0,1\}$ be a decision variable that indicates whether an impression from campaign $k \in \Gamma(v,i)$ is put in pattern slot $\ell \in \{1...L_v\}$, and let $\bar{z}_{k\ell} = \sum_{\ell'=1}^{\ell} z_{k\ell'}$ be the cumulative number of times that campaign k appears in the first ℓ slots. For the f_k impressions of campaign k to be spread exactly uniformly the across L_v slots, we need the cumulative count $\bar{z}_{k\ell}$ to grow at a constant rate f_k/L_v, i.e., by the time we reach slot ℓ of the pattern, $\bar{z}_{k\ell}$ should equal the target cumulative count $T_\ell = \frac{f_k}{L_v}\ell$. Kubiak and Sethi (1991) quadratically penalize the deviation between $\bar{z}_{k\ell}$ and the target cumulative count T_ℓ. For any fixed $\mathbf{b}$, the following math program, with decision variables $z_{k\ell}$, produces a maximally-

paced pattern by minimizing non-uniformity as measured by Kubiak and Sethi:

$$\pi_{pacing}(\mathbf{b}) \;=\; \text{Minimize} \quad \sum_{k\in\Gamma(v,i)} \sum_{\ell=1}^{L_v} \left(\sum_{\ell'=1}^{\ell} z_{k\ell'} - b_k T_\ell \right)^2 \tag{8a}$$

$$\sum_{\ell=1}^{L_v} z_{k\ell} \;=\; b_k f_k \qquad \forall k \in \Gamma(v,i) \tag{8b}$$

$$\sum_{k\in\Gamma(v,i)} z_{k\ell} \;\leq\; 1 \qquad \forall \ell = 1, ..., L_v \tag{8c}$$

$$z_{k\ell} \in \{0,1\} \tag{8d}$$

Constraint (8b) ensures we include exactly f_k impressions of campaign k if campaign k is supposed to be in the pattern (i.e., $b_k = 1$), and zero impressions otherwise. Constraint (8c) ensures that each slot in the pattern is occupied by at most one campaign. The target cumulative count T_ℓ in the objective is multiplied by b_k to ensure we only penalize non-uniform pacing for campaigns that are in the pattern (when $b_k = 0$, all $z_{k\ell}$'s are zero thanks to constraint (8b)).

Kubiak and Sethi (1994) show that this quadratic program can be transformed in polynomial time into an *assignment problem*, i.e., a weighted bipartite matching, with $\sum_{k\in\Gamma(v,i)} f_k$ supply nodes and L_v demand nodes. Assignment problems are fundamental to combinatorial optimization and network flow theory for which many efficient solution techniques are available, e.g., the best implementation of the Hungarian Algorithms has $O\big(L_v^3\big)$ runtime (see Ahuja et al., 1993, Ch.12). However, in our case, we are not interested in solving (8) in isolation but rather we wish to solve (8) as an inner-optimization within (PG). Unfortunately, we cannot transform (8) into an assignment problem when the $\mathbf{b}$ vector is also a decision variable. Instead, to integrate (8) into (PG), we use (8a) as the objective and include the constraints (8b,c,d) in (PG). This adds $O(L_v|\Gamma(v,i)|)$ binary variables and $O(L_v + |\Gamma(v,i)|)$ constraints to (PG). Using CPLEX, solving each instance of this extended formulation, which is a quadratic mixed integer program, takes only a few seconds. This is slower than solving a binary knapsack problem via dynamic programming (as we do when $\pi(\mathbf{b})$ is linear), but it is important to note that (PA) and (PG) are solved independently for each supply node (v,i), and can be run in parallel across many machines. So, the additional runtime of (PG) can be compensated for by using more parallel computing nodes. The runtime of a few seconds for (PG) is within practical limits given that large publishers in industry have thousands of computing nodes at their disposal.

One possible limitation to Kubiak's model (8) is that the target cumulative curve for each and every campaign, $T_\ell = \frac{f_k}{L_v}\ell$, starts from time zero (i.e., the first slot in the pattern). One could modify the model by introducing additional variables, I_k, which allow the math program to decide from which slot the target cumulative curve starts, making the target curve $T_\ell = \big(\frac{f_k}{L_v}\ell - I_k\big)^+$. Alternatively, the publisher can fix the starting points I_k as parameters using historical exposure time, to provide continuity of pacing from one planning period to the next. In either case, the runtime of (PG) in extended form is not appreciably affected by these modifications. In fact, the target cumulative count T_ℓ can be defined as any general function of ℓ to achieve any desired pacing

pattern. Another useful case is $T_\ell = \frac{f_k}{L_v} t_\ell$, where the parameter t_ℓ is the anticipated arrival time of the user's ℓ^{th} visit. If the approximate timing of user visits can be forecasted by the publisher, then we can construct patterns that deliver ads uniformly across time, as opposed to across serving opportunities.

A more recent, but more complex, model is due to Bollapragada et al. (2004) who consider the problem of uniformly arranging TV advertisements across commercial breaks. They formalize the problem as arranging f_k balls of different colors, indexed by k, into L_v slots ($\sum_k f_k \leq L_v$) such that balls of the same color are as evenly spaced as possible. In their model, the space between any two consecutive balls of the same color k is expected to be L_v/f_k. Any deviation from this distance is penalized linearly in the objective. Let the binary variable $z_{j_k \ell}$ model whether the j^{th} impression of campaign k is placed in slot ℓ of the pattern, and let $Z_{j_k} = \sum_{\ell=1}^{L_v} \ell z_{j_k \ell}$ be the slot number in which the j^{th} impression of campaign k appears. Using Bollapragada's model, our inner optimization problem is defined as:

$$\pi_{pacing}(\mathbf{b}) \;=\; \text{Minimize} \quad \sum_k \sum_{j_k=2}^{f_k} \left| Z_{j_k} - Z_{(j-1)_k} - \frac{L_v}{f_k} b_k \right| \tag{9a}$$

$$\sum_{j_k=1}^{f_k} \sum_{\ell=1}^{L_v} z_{j_k \ell} \;=\; f_k b_k \qquad \forall k \tag{9b}$$

$$\sum_k \sum_{j_k=1}^{f_k} z_{j_k \ell} \;\leq\; 1 \qquad \forall \ell = 1, ..., L_v \tag{9c}$$

$$Z_{j_k} \;=\; \sum_{\ell=1}^{L_v} \ell z_{j_k \ell} \qquad \forall k,\; j_k = 1, ..., f_k \tag{9d}$$

$$Z_{j_k} \;\geq\; Z_{(j-1)_k} + 1 \qquad \forall k,\; j_k = 2, ..., f_k \tag{9e}$$

$$z_{j_k \ell} \in \{0, 1\}, \quad Z_{j_k} : \text{Integers} \tag{9f}$$

Constraints (9b) and (9c) perform the same function as (8b) and (8c). Constraint (9d) establishes the relationship between variables $z_{j_k \ell}$ and Z_{j_k}, and constraint (9e) ensures that the j^{th} impression of campaign k is placed after the $(j-1)^{th}$ impression. Bollapragada et al. (2004) show that this problem can be cast as a minimum-cost network flow problem which is somewhat faster to solve than the integer program (9), but not appreciably faster due to the exponential number of arcs in the resulting network graph. The authors then develop a customized branch-and-bound algorithm and propose many heuristics for obtaining good solutions in reasonable time. In a subsequent paper, Brusco (2008) develops an enhanced branch-and-bound algorithm for (9) as well as a simulated annealing heuristic that also handles more general L_p-norm penalty functions.

In the extended formulation of subproblem (PG) which incorporates Bollapragada's (9a) as the objective and (9b-f) as constraints, there are $O(L_v \sum_{k \in \Gamma(v,i)} f_k)$ additional binary variables and $O(L_v + \sum_{k \in \Gamma(v,i)} f_k)$ additional constraints. From our experience, Bollapragada's model results in much slower (and less predictable) runtimes than Kubiak's. Qualitatively speaking, the uniformity of patterns produced by one model does not exhibit any obvious visual advantage over the other. This suggests that for the goal of maximally pacing ads, one should prefer to extend (PG) using

(8) rather than (9).

4. Competing campaigns

Campaigns of competing brands may target similar user demographics, and such advertisers may wish to stop their audience from being exposed to their competition's ads. For any set of competing campaigns $C \subseteq \mathcal{K}$, the publisher can include a constraint of the form $\sum_{k \in C} b_k \leq 1$ in (PG) so at most one of the competing campaigns is included in the pattern. Such constraints are well-known in the integer programming literature as SOS1 constraints, for which effective methods are known and embedded into integer programming solvers.

Final Remarks

One may also consider a weighted combination of multiple measures:

$$\pi(\mathbf{b}) \;=\; \lambda_1 \pi_{pacing}(\mathbf{b}) \;+\; \lambda_2 \pi_{diversity}(\mathbf{b}) \;+\; \lambda_3 \pi_{excess}(\mathbf{b}).$$

Furthermore, to maintain linearity of $\pi(\mathbf{b})$ which speeds up the solution time of (PG), the publisher may exclude the pacing term from $\pi(\mathbf{b})$ to maintain the knapsack structure of (PG), and instead use one of the quick greedy heuristics proposed by Bollapragada et al. (2004) as a post-processing step to rearrange the impressions within the generated patterns.

C Multiple Ad Positions and Two-dimensional Patterns

Throughout the paper we assume the publisher's webpage has a single advertising position, where an ad can be shown. Therefore, our patterns are designed to deliver a single ad impression upon each user visit. In this section we discuss the changes to our model that apply when the publisher's page has multiple ad positions. This involves creating patterns that are two-dimensional. Each column in the pattern holds the ads that are shown simultaneously to a user upon a single visit. For instance, Figure 2 can be viewed as a 3×8 pattern. On the first visit, campaign A is shown in all three ad positions of the webpage; for the second visit, the user is shown campaign C in position 1, and campaign B in both positions 2 and 3; and so on.

Before we discuss how two-dimensional patterns can be constructed, we would like to point out many practical cases in which one-dimensional patterns are still appropriate even when the webpage has multiple ad positions. We use $h = 1, ..., H$ to index the ad positions.

1. *When ad positions are different and sold separately to advertisers*: For example, each ad campaign uses a specific size of graphic that is designed for a specific position on the page which the advertiser has booked (e.g., the wide banner ad on the top, or the tall skyscraper ad on the right side of the page). In this case, the publisher's ad allocation problem decomposes by ad position. The publisher needs to solve H separate problems and maintain a separate pattern pool $\mathcal{P}_{vih}$

for each user type (v, i) and each ad position h. Upon a user's first visit, s/he is assigned to H patterns, independently sampled from the optimal solutions obtained for each ad position.

2. *When advertisers do not strictly require the frequency to be delivered across separate user visits*: In this case, showing multiple instances of the same campaign in different ad positions upon a single visit will count toward the frequency requirement. To model this case, we simply create one-dimensional patterns of length HL_v, and we use H impressions at a time, upon each user visit. Note that if the pattern quality measure includes a pacing cost function (π_{pacing}), impressions of the same campaign will be well-spread throughout the pattern, making it unlikely for the same ad to appear in multiple positions on the page (see Appendix B for a discussion of how we implement π_{pacing}). The pacing model of Bollapragada et al. (2004) will try to arrange a campaign so that consecutive impressions are $HL_v/f_k > H$ slots apart. In the pacing model of Kubiak and Sethi (1991), as discussed in Appendix B, we can assign arrival times t_ℓ to pattern slots such that the first H slots in the pattern are assigned $t_\ell = 1$, the following H slots are all assigned $t_\ell = 2$, and so on. This will more significantly discourage multiple instances of the same campaign from appearing in multiple ad positions on the page.

3. *Newsfeed ads, video ads, and dynamic webpages*: Many of modern webpages are designed in a dynamic fashion so that the delineation of when a page loads, or when a user navigates from one page to another is less clear. For instance, the banner ad in Yahoo Mail is reloaded with a new ad every time the user scrolls down for at least 1 page through the email list. Similarly, ads on Facebook (and many websites with native advertising) load within the news feed as the user scrolls down the page. Video ads, which are the fastest growing segment of online advertising, also demonstrate the same behavior. A sequence of video ads can be shown to the user during a long movie (similar to commercial breaks on TV), or multiple banner ads can be overlaid on a video clip at different points in time (common practice on YouTube). Finally, most ads served through Google AdSense are automatically reloaded with new advertising every 20-30 seconds. In all these cases, a one-dimensional pattern is appropriate for serving ads, especially since the number of ads required is not known beforehand and depends on the amount of user interaction (scrolling action or time spent on the page).

If none of the above conditions are met, we propose the use of two-dimensional patterns. The only changes to our mathematical framework will be a division by H in the left-hand side of constraint (3c), and a reformulation of (PG) so it constructs two-dimensional patterns. As before, assume the pattern has length L_v with columns indexed by ℓ which correspond to the number of visits made by a type-v user. The pattern also has a height H with rows indexed by h, which correspond to the number of positions on the webpage. Upon the user's ℓ^{th} visit, all H slots in the ℓ^{th} column of the pattern appear in the corresponding H ad positions on the webpage, and therefore, are seen by the user at the same time.

Let the binary variable b_{kh} denote whether campaign k is included in row h of the pattern. Note that $b_{kh} = 1$ implies all f_k impressions of k appear in ad position h on the webpage. However,

once a solution b^*_{kh} is found, the publisher can shuffle the ads within the pattern column (i.e., across ad positions on the page) without affecting any of the pattern quality metrics discussed in Appendix B. Sub-problem (PG) can be cast as:

$$\text{Minimize} \quad \pi(\mathbf{b}) \; - \sum_{k \in \Gamma(v,i)} \bar{\alpha}^*_{vik} b_k \tag{10a}$$

$$\text{s.t.} \quad \sum_{k \in \Gamma(v,i)} f_k b_{kh} \leq L_v \qquad \forall h = 1, ..., H \tag{10b}$$

$$b_k \equiv \sum_{h=1}^{H} b_{kh} \leq 1 \qquad \forall k \in \Gamma(v,i) \tag{10c}$$

$$b_{kh} \in \{0, 1\}, \qquad \forall k \in \Gamma(v,i), \; \forall h = 1, ..., H \tag{10d}$$

Constraint (10b) is analogous to (5b) and ensures each row of the pattern is filled with at most L_v impressions. As we discussed above, the publisher would only use two-dimensional patterns when showing multiple impressions of the same ad upon a single visit does not count toward the frequency requirement of the campaign. Constraint (10c) serves to ensure that a campaign is not assigned to more than one ad position. It also implies that the campaign does not appear more than once throughout the pattern.

It is straightforward to see how the cost functions from Appendix B can be adapted to two-dimensional patterns. We would use $\pi_{excess}(\mathbf{b}) = (HL_v - \sum_k f_k b_k)\bar{c}_{vi}$. The diversity cost measure $\pi_{diversity}(\mathbf{b})$ stays unchanged, and the pacing cost function $\pi_{pacing}(\mathbf{b})$ decomposes into separate inner-optimization problems for each row of the pattern (i.e., each ad position on the page).

If the cost function $\pi(\mathbf{b})$ is linear in b_k (as it is, when pattern quality is measured by excess and/or diversity), then (10) becomes an instance of a binary *multiple knapsack problem*. This problem is known to be NP-hard for which dynamic programming is no longer an efficient pseudo-polynomial solution technique. Appropriate algorithms for multiple knapsack problems are discussed in Martello and Toth (1990, Ch.6).

D Modeling Random Arrivals

A core assumption in our methodology of serving ads using predefined patterns that span across time is that each user visits the publisher's website at least as many times as the number of slots in his/her assigned pattern. Otherwise, the pattern will not be delivered completely and the campaigns which do not hit their target frequency will not "reach" that user as planned in the optimization model. We suggested earlier in §4 that the publisher may cluster users based on browsing behavior, such that all users of the same visit-type v have the same probability distribution $\phi_v(\cdot)$ for the number of visits over the planning period. Recall that we defined pattern lengths as $L_v = \Phi_v^{-1}(\varepsilon)$, where $1-\varepsilon$ was the desired minimum probability that the user of type v makes at least L_v visits and views the whole pattern. However, this approach may be overly conservative and exclude a significant portion of the publisher's traffic from being used for R&F campaigns. For instance, if the number of visits from a particular user type follows a Poisson distribution with rate parameter 30 (over the

planning horizon), we can only plan for 20 visits from the user if we aim for 95% assurance that the user fully sees the pattern. Therefore, on average 10 visits ($E[\max(0, X - 20)] = 10.049$ when $X \sim Poiss(30)$), i.e., 1/3 of the impression traffic from this user type is not considered for R&F planning. In this section we develop a probabilistic pattern generation mechanism that explicitly incorporates the visit frequency distribution of users. We follow with numerical experiments that illustrate the significant improvement in the utilization of supply and reducing under-delivery when our probabilistic model is employed. This comes at a price, however, since the pattern-generating sub-problem becomes more complex and thus harder to solve.

Let $\phi_v^{(\ell)}$ denote the probability that a user of visit-type v makes exactly $\ell \in \{1, \ldots, \bar{L}_v\}$ visits. Parameter $\bar{L}_v$ models the *maximum* number of visits ever expected from a type-v user and is greater than the *anticipated* number of visits, L_v, which occurs with a high probability $1-\varepsilon$. To prepare for all possible number of visits from the user, we now consider designing patterns of the full length $\bar{L}_v$. As before, we use the binary variables b_k to denote whether campaign k is included in the pattern. For each slot $\ell = \{1, \ldots, \bar{L}_v\}$ in the pattern, let $z_{k\ell} \in \{0, 1\}$ denote whether the slot is occupied by campaign k, and let $\bar{z}_{k\ell} = \sum_{\ell'=1}^{\ell} z_{k\ell'}$ denote the cumulative number of times campaign k appears in the first ℓ slots. Binary indicator variable $I_{k\ell}$ measures whether or not all f_k impressions of campaign k are positioned in the first ℓ slots. That is, $I_{k\ell} = 0$ if $\bar{z}_{k\ell} < f_k$ and $I_{k\ell} = 1$ as soon as $\bar{z}_{k\ell} = f_k$.

Note that $\bar{b}_{kp} = \sum_{k=1}^{\bar{L}_v} \phi_v^{(\ell)} I_{k\ell}$ gives the probability that campaign k will reach its frequency requirement f_k on a type-v user, should s/he be assigned pattern p. For each campaign k, we have a binomial process, where we make y_{vip} trials (user assignments of the pattern), each having a success (reach) probability of $\bar{b}_{kp}$. Thus, $\sum_n \bar{b}_{kp} y_{vip}$ gives the expected number of times that k is reached within user class (v, i). The pattern assignment problem (PA) becomes:

$$\text{(PA-R):} \quad \Psi_{vi}^{(R)} := \text{Minimize} \quad \sum_{p \in \mathcal{P}_{vi}} \pi_{vip} y_{vip} \qquad \underline{\text{Duals:}} \tag{11a}$$

$$\sum_{p \in \mathcal{P}_{vi}} \bar{b}_{kp} y_{vip} = s_{vi} x_{vik}^* \quad \forall k \in \Gamma(v, i) \qquad \bar{\alpha}_{vik}^{(R)} \text{ (free)} \tag{11b}$$

$$\sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq s_{vi} \qquad \bar{\beta}_{vi}^{(R)} \geq 0 \tag{11c}$$

$$y_{vip} \geq 0 \quad \forall p \in \mathcal{P}_{vi} \qquad - \tag{11d}$$

where the optimal reach proportions x_{vik}^* from (RA-δ) are sought in expectation. The only change from (PA) is the substitution of b_{kp} from (4b) with $\bar{b}_{kp}$ in (11b). The pattern generating subproblem takes the following form:

$$\text{(PG-R):} \qquad \psi_{vi}^{(R)} := \text{Maximize} \quad \sum_{k\in\Gamma(v,i)} \bar{\alpha}_{vik}^{*(R)} \underbrace{\Big(\sum_{k=1}^{\bar{L}_v} \phi_v^{(\ell)} I_{k\ell}\Big)}_{\bar{b}_k} - \pi(\mathbf{b}) \tag{12a}$$

$$\sum_{k\in\Gamma(v,i)} z_{k\ell} \le 1 \qquad \ell = 1, ..., \bar{L}_v \tag{12b}$$

$$\sum_{\ell=1}^{\bar{L}_v} z_{k\ell} = f_k b_k \qquad \forall k \in \Gamma(v,i) \tag{12c}$$

$$\sum_{\ell'=1}^{k} z_{k\ell'} \le f_k - 1 + I_{k\ell} \qquad \forall k \in \Gamma(v,i),\ \ell = 1, ..., \bar{L}_v \tag{12d}$$

$$\sum_{\ell'=1}^{k} z_{k\ell'} \ge f_k I_{k\ell} \qquad \forall k \in \Gamma(v,i),\ \ell = 1, ..., \bar{L}_v \tag{12e}$$

$$b_k, z_{k\ell}, I_{k\ell} \in \{0,1\} \tag{12f}$$

The first set of constraints (12b) ensure that at most one campaign occupies each slot. The second set of constraints (12c) require each campaign k to appear exactly f_k times throughout the pattern if we choose to include k in the pattern ($b_k = 1$), and zero otherwise (if $b_k = 0$). The left-hand side in (12d) and (12e) are the cumulative impression counts $\bar{z}_{k\ell}$. Constraints (12d) enforce $I_{k\ell} = 1$ when $\bar{z}_{k\ell} = f_k$, whereas constraints (12e) enforce $I_{k\ell} = 0$ if $\bar{z}_{k\ell} < f_k$. The above binary program has $O(\bar{L}_v|\Gamma(v,i)|)$ variables and constraints. As soon as $\psi_{vi}^{*(R)} + \bar{\beta}_{vi}^{*(R)} \ge 0$, the optimal solution to (PA-R) has been found. Otherwise, we add the pattern constructed by (PG-R) to $\mathcal{P}_{vi}$ with reach probability parameters $\bar{b}_{kp} = \sum_{k=1}^{\bar{L}_v} \phi_v^{(\ell)} I_{k\ell}$ and re-solve (PA-R) to obtain new dual values $\bar{\alpha}_{vik}^{*(R)}$ and $\bar{\beta}_{vi}^{*(R)}$. Again, for possible functional choices for $\pi(\mathbf{b})$, we refer the reader to Appendix B.

When no pattern quality measure is used, or during feasibility phase of Pattern-HCG when $\pi(\mathbf{b})$ is non-existent, it is easy to show that the optimal solution always places all f_k impressions of each campaign in successive slots. This is due to the fact that every deviation from such structure will only decrease the chance of (at least) one campaign from being fully observed by the user, $\bar{b}_k$, and therefore worsens the objective value (12a).

Computational Experiments:

In this section, we examine how efficiently the random supply of impressions (coming from a random number of arrivals per user) can be allocated using our probabilistic model, compared to our deterministic model of §5, and how this affects under-delivery and non-representativeness.

For efficiently solving the binary integer subproblem (PA-R), we used CPLEX 12.6 API for Matlab® and due to compatibility issues we could no longer take advantage of parallelization and so conducting the test on Yahoo data was impractical. Instead, we created a small synthetic graph with roughly 500 supply nodes and 30 demand nodes. In each supply node, we assumed three user visit-types $\mathcal{V} = \{\text{low,med,high}\}$ whose number of visits follows a Poisson distribution

Visiting Rates (Poisson)	Random Arrival Pattern Lengths	Deterministic Pattern Finish Probability $(1-\varepsilon)$	Under-delivery		Non-representat.	
			Det.	Rand.	Det.	Rand.
$\lambda = \{8.7, 18, 27\}$	$\bar{L} = \{20, 35, 45\}$	25%	0.255	0.085	245.9	305.1
$\lambda = \{11, 21, 31\}$	$\bar{L} = \{25, 40, 50\}$	50%	0.174	0.043	259.6	189.8
$\lambda = \{14, 25, 36\}$	$\bar{L} = \{30, 45, 55\}$	80%	0.138	0.034	271.8	125.4
$\lambda = \{16, 28, 39\}$	$\bar{L} = \{35, 45, 60\}$	90%	0.123	0.032	266.8	116.3
$\lambda = \{17, 30, 41\}$	$\bar{L} = \{35, 50, 65\}$	95%	0.113	0.030	276.2	111.9

Table 1: Test cases and results under random arrival scenario. Deterministic pattern lengths are set to $L = \{10, 20, 30\}$ in all cases.

at different rates, specified by the vector $\lambda = \{\lambda_{\text{low}}, \lambda_{\text{med}}, \lambda_{\text{high}}\}$. Deterministic pattern lengths, $L = \{L_{\text{low}}, L_{\text{med}}, L_{\text{high}}\}$, employed by our model are fixed at $\{10, 20, 30\}$ and we vary the arrival rate parameters λ_v so that the probability of each type-v user visiting at least L_v times is set close to a desired threshold (see the third column in Table 1). For example, Poisson random variables with mean parameters $\lambda = \{8.7, 18, 27\}$ all have about a 25% chance of exceeding $\{10, 20, 30\}$, respectively. The pattern lengths for the random arrival model, $\bar{L}_v$ (second column in Table 1) are chosen to cover at least 99% of the support of the corresponding Poisson distribution (e.g., looking at the first row in Table 1, Poisson random variables with rates $\lambda = \{8.7, 18, 27\}$ have only a 0.001 chance of exceeding $\bar{L} = \{20, 35, 45\}$, respectively).

We specifically generated our synthetic instance such that the supply of users is enough to satisfy the reach requirements from all campaigns. Therefore, the only factor that may cause under-delivery is whether or not users make enough visits for the frequency requirements to be met. The quality of the solution depends highly on how well the f_k impressions of each campaign are arranged into the slots of a pattern so it is robust to truncation. Our probabilistic model explicitly takes into account the user visit distribution $\phi_v(\cdot)$ when constructing patterns. For our comparison to be conservative, in our deterministic solution of §5, we moved all excess impressions to the end of every pattern, and positioned all impressions of the same campaign sequentially. The orders of different campaigns in the patterns were selected purely at random.

Our experiments, shown in Table 1, demonstrate a significant improvement in performance when our probabilistic model is employed. Note that the random arrival model also provides a structural advantage over the deterministic model: Since pattern lengths $\bar{L}_v$ are higher than that of L_v, campaigns with high f_k may fit into $\bar{L}_v$ but not L_v for low-visiting types v. Therefore, the connectivity of each supply node $|\Gamma(v, i)|$ is larger in the probabilistic model. Note that when users of all visit types are expected to complete L_v visits with 95% chance (last row in Table 1), we observe almost no under-delivery (3%) using our probabilistic solution, whereas the deterministic solution yields 11% under-delivery due to under-utilizing the (quite ample) impression supply. Note that in this case, for low-visiting users with average visit frequency of $\lambda_{\text{low}} = 17$, our deterministic and probabilistic models use pattern lengths of $L_{\text{low}} = 10$ (too low) and $\bar{L}_{\text{low}} = 35$, respectively.

E Monolithic Formulation of the R&F Planning Problem

In §5, we enumerated a number of practical issues with our earlier model presented in the conference paper Hojjat et al. (2014). In this section we elaborate on some of those deficiences, in particular the inability of our model from Hojjat et al. (2014) to uniquely characterize the primal solution as a function of the dual solution. For convenient reference, we present our earlier model using the notation in this manuscript, and derive some additional properties of that model which were not discussed previously. The following math program, translated from Hojjat et al. (2014), combines reach allocation and pattern assignment into a single "monolithic" component, and has decision variables x_{vik}, u_k, and y_{vip}:

$$\text{Minimize} \quad \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi} w_k}{2\theta_k} (x_{vik} - \theta_k)^2 + \sum_k c_k u_k + \sum_{v,i} \sum_{p\in\mathcal{P}_{vi}} \pi_{vip} y_{vip} \tag{13a}$$

$$\text{s.t.} \quad x_{vik} = \frac{1}{s_{vi}} \sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip} \qquad \forall v, i, k \in \Gamma(v,i) \tag{13b}$$

$$\sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} + u_k \geq r_k \qquad \forall k \tag{13c}$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq 1 \qquad \forall v, i \tag{13d}$$

$$\sum_{p\in\mathcal{P}_{vi}} y_{vip} \leq s_{vi} \qquad \forall v, i \tag{13e}$$

$$0 \leq x_{vik} \leq 1 \tag{13f}$$

$$y_{vip} \geq 0,\ u_k \geq 0 \tag{13g}$$

The solution assigns a x_{vik}-fraction of the users of type (v,i) to campaign k, falls short of campaign k's reach target by u_k users, and assigns pattern p to users of type (v,i) exactly y_{vip} times. The objective combines both aggregate and disaggregate quality metrics into one composite function. The first two terms reflect the aggregate quality metric used within this paper, i.e., by minimizing non-representativeness and under-delivery. The third term reflects disaggregate quality, i.e., by minimizing the total cost of selected patterns. As in this paper, w_k is the weight given to the non-representativeness term which quadratically penalizes deviations from the perfectly-representative solution, i.e., one that assigns campaign k a $\theta_k = r_k / \sum_{(v,i)\in\Gamma(k)} s_{vi}$ proportion of (v,i)-users. Under-delivery u_k is penalized at the marginal cost c_k, and pattern p has cost π_{vip} when assigned to a user of type (v,i).

Constraint (13b) links the reach allocation variable x_{vik} to the pattern assignment variables y_{vip}, and can be viewed as a summary statistic of pattern assignment that indicates what proportion of type-(v,i) users are reached by campaign k. Note that the parameter b_{kp} is 1 if campaign k is in pattern p, and 0 otherwise. Constraints (13c) and (13d) are supply and demand constraints from (RA). Constraint (13e) ensures that the total number of patterns assigned to (v,i)-users does not exceed the number of unique users available (recall each user is assigned a single pattern). Constraints (13f) and (13g) provide bounds on the variables. Although x_{vik} represents a proportion, as we argued in §5.1, we do not need constraints of the form $\sum_{k\in\Gamma(v,i)} x_{vik} \leq 1$ because a user can

be reached by more than one campaign as long as the pattern length L_v is sufficiently large.

We now show that a number of structural properties hold, which allows us to simplify the above formulation. We begin by pointing out that the upper bound in constraint (13f) is redundant. To see this, note that for any given user type (v, i) we have:

$$x_{vik} = \frac{1}{s_{vi}} \sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip} \leq \frac{1}{s_{vi}} \sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq 1.$$

The first equality follows by definition of constraint (13b). The next inequality follows since each b_{kp} value is at most 1. Finally, the last inequality follows from constraint (13e).

Next, we show that the user-based supply constraint (13e) is always tighter than the impression-based supply constraint (13d). In other words, (13d) is dominated by (13e), making (13d) redundant. To see this, note that for any given user type (v, i) we have:

$$\sum_{k \in \Gamma(v,i)} \frac{f_k}{L_v} x_{vik} = \sum_{k \in \Gamma(v,i)} \frac{f_k}{L_v} \left(\frac{1}{s_{vi}} \sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip} \right) = \frac{1}{s_{vi}} \sum_{p \in \mathcal{P}_{vi}} \left(\frac{\sum_{k \in \Gamma(v,i)} f_k b_{kp}}{L_v} \right) y_{vip} \leq \frac{1}{s_{vi}} \sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq 1.$$

The first equality follows by definition of constraint (13b). The second equality is a simple rearrangement of terms. The next inequality is due to fact that a pattern assigned to a type-(v, i) user has L_v slots, and since reaching each campaign k occupies f_k slots, $\sum_{k \in \Gamma(v,i)} f_k b_{kp} \leq L_v$ must always hold for any pattern $p \in \mathcal{P}_{vi}$. The last inequality follows from constraint (13e).

Finally, after dropping the redundant constraints (13d) and (13f) and eliminating x_{vik} by substitution using constraint (13b), we can represent the monolithic formulation of Hojjat et al. (2014) in the following simplified form:

$$\begin{aligned}
\text{(FP):} \quad \text{Minimize} \quad & \sum_k \sum_{(v,i) \in \Gamma(k)} \frac{s_{vi} w_k}{2\theta_k} \left(\sum_{p \in \mathcal{P}_{vi}} \frac{b_{kp}}{s_{vi}} y_{vip} - \theta_k \right)^2 + \sum_k c_k u_k + \sum_{v,i} \sum_{p \in \mathcal{P}_{vi}} \pi_{vip} y_{vip} && \underline{\text{Duals}}(\text{All} \geq 0) \\
& \sum_{(v,i) \in \Gamma(k)} \sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip} + u_k \geq r_k \qquad \forall k && \alpha_k \\
& \sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq s_{vi} \qquad \forall v, i && \beta_{vi} \\
& y_{vip} \geq 0, \; u_k \geq 0 && \gamma_{vip}, \varphi_k
\end{aligned}$$

The Lagrangean of problem (FP) is:

$$\begin{aligned}
\mathcal{L} = & \sum_k \sum_{(v,i) \in \Gamma(k)} \frac{s_{vi} w_k}{2\theta_k} \left(\sum_{p \in \mathcal{P}_{vi}} \frac{b_{kp}}{s_{vi}} y_{vip} - \theta_k \right)^2 + \sum_k c_k u_k + \sum_{v,i} \sum_{p \in \mathcal{P}_{vi}} \pi_{vip} y_{vip} \\
& + \sum_k \alpha_k \left(r_k - \sum_{(v,i) \in \Gamma(k)} \sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip} - u_k \right) + \sum_{v,i} \beta_{vi} \left(\sum_{p \in \mathcal{P}_{vi}} y_{vip} - s_{vi} \right) - \sum_{v,i} \sum_{p \in \mathcal{P}_{vi}} \gamma_{vip} y_{vip} - \sum_k \varphi_k u_k
\end{aligned}$$

The stationarity condition $\frac{\partial \mathcal{L}}{\partial y_{vip}} = 0$ yields the reduced cost function for the variable y_{vip}:

$$\gamma_{vip} \;=\; \sum_{k \in \Gamma(v,i)} \left(\frac{w_k}{\theta_k s_{vi}} \sum_{p' \in \mathcal{P}_{vi}} b_{kp'} y_{vip'} - w_k - \alpha_k \right) b_{kp} + \pi_{vip} + \beta_{vi}. \tag{14}$$

An immediate and important observation is that the stationarity condition does not establish a mapping from the dual variables α_k and β_{vi} to a unique solution for the primal variable y_{vip}; i.e., we cannot rearrange (14) in a way that isolates y_{vip} as a function of α_k and β_{vi}. In contrast, Theorem 1 shows that such a mapping from the dual variables to a unique primal solution exists for this paper's (RA-δ) formulation. Consequently, the Modified SHALE method, which we use to efficiently solve (RA-δ) in a parallelized manner, cannot be applied to (FP). Moreover, even after making the substitution $x_{vik} = \frac{1}{s_{vi}} \sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip}$ to recover the reach allocation using constraint (13b), the reduced cost function (14) simplifies to

$$\gamma_{vip} \;=\; \sum_{k \in \Gamma(v,i)} \left(\frac{w_k}{\theta_k} x_{vik} - w_k - \alpha_k \right) b_{kp} + \pi_{vip} + \beta_{vi},$$

which still does not admit a mapping from the dual variables α_k and β_{vi} to a unique solution for the primal variable x_{vik}. Consequently, even if we could solve (FP) efficiently, its solution is not generalizable in the way that the solution to (RA-δ) is. These structural limitations greatly diminish the attractiveness of solving (FP) using column generation in practice.

For completeness, we conclude this section by deriving the pattern generating problem corresponding to (FP), and describe how column generation can in theory be used to solve (FP). At a high level, the idea is to start with a small pool of patterns, solve (FP), and then use the current optimal primal/dual solution as feedback to construct new patterns which can improve the current solution. We then add these improving patterns to our pattern pools $\mathcal{P}_{vi}$ and solve (FP) again, repeating this procedure until no improving pattern can be constructed.

Given a primal/dual solution $\{y^*_{vip}, \alpha^*_k, \beta^*_{vi}\}$ to (FP), the following pattern generating problem finds a pattern with minimum reduced cost:

$$\begin{aligned} \text{(FPS)} \qquad \psi_{vi} := \text{Minimize} \quad & \pi(\mathbf{b}) \;+\; \sum_{k \in \Gamma(v,i)} \left(\frac{w_k}{\theta_k s_{vi}} \sum_{p \in \mathcal{P}_{vi}} b_{kp} y^*_{vip} - w_k - \alpha^*_k \right) b_k \\ \text{s.t.} \quad & \sum_{k \in \Gamma(v,i)} f_k b_k \le L_v \\ & b_k \in \{0,1\}, \quad \forall k \in \Gamma(v,i) \end{aligned}$$

The variables here are b_k, not to be confused with the parameters b_{kp} which remain constant. We use $\mathbf{b} = \{b_k : k \in \Gamma(v,i)\}$ to denote the vector of all decision variables. Recall that several pattern cost functions $\pi(\mathbf{b})$ were introduced in Appendix B.

If $\psi^*_{vi} + \beta^*_{vi} < 0$ for any supply node (v,i), it is beneficial to add the new pattern p' to $\mathcal{P}_{vi}$ with $b_{kp'} = b^*_k$ and $\pi_{vip'} = \pi(\mathbf{b}^*)$, and the solution to (FP) will be improved. On the other hand, if

$\psi^*_{vi} + \beta^*_{vi} \geq 0$ for all (v,i), the solution to (FP) is optimal. To initialize the pattern pools $\mathcal{P}_{vi}$, one can initially solve (FPS) with $\alpha_k = \beta_{vi} = y_{vip} = 0$, which is primal/dual feasible.

The column generation scheme which alternates between solving (FP) and (FPS) is quite slow. Due to a lack of generalizability, we cannot use the efficient SHALE algorithm to solve (FP), which needs to be solved multiple times. Moreover, the pattern cost $\pi(\mathbf{b})$ is always a part of the objectives of (FPS) and (FP). Although (FPS) parallelizes by supply node (v,i), solving (FPS) can still be computationally expensive when $\pi(\mathbf{b})$ is nonlinear in the b_k variables, e.g., when $\pi(\mathbf{b})$ measures user-level pacing (see Appendix B). In contrast, our Pattern-HCG algorithm has a *feasibility* phase followed by a *pattern improvement* phase. During the feasibility phase, we iterate between solving (RA-δ) efficiently using Modified-SHALE and generating and assigning patterns using (PG-F) and (PA-F). Not only do (PG-F) and (PA-F) parallelize by (v,i), but (PG-F) is a binary knapsack problem that is independent of $\pi(\mathbf{b})$ and is very quick to solve. Finally, during the pattern-improvement phase, we no longer need to solve (RA-δ), and pattern assignment (PA) and generation (PG), which now involve the $\pi(\mathbf{b})$ metric, converge quickly since they are both parallelized by supply node (v,i) and do not need to interact with the variables from (RA-δ). In summary, for a number of structural reasons, Pattern-HCG is much more efficient than the standard implementation of column generation applied to the monolithic formulation presented in this section. In essence, because pattern generation and pattern assignment components must alternate in a column generation scheme, and reach allocation and pattern assignment are merged together into one component in (FP), the pattern assignment step is bogged down by needing to be re-solved with the large math program that constitutes the reach allocation. Our Pattern-HCG decouples reach allocation from pattern assignment, allowing each of these components to be solved efficiently.

F Proof of Theorem 1 (Generalizability of RA-δ)

Theorem. *The optimal primal and dual solutions of (RA-δ) satisfy the following relationships:*

1. *The optimal primal solution x^*_{vik} can be computed from the optimal dual solution $\{\alpha^*_k, \beta^*_{vi}\}$, and is given by: $x^*_{vik} = g_{vik}(\alpha^*_k, \beta^*_{vi}) \equiv \min\Big[1, \max\Big[0, \theta_k + \frac{\theta_k}{w_k}\big(\alpha^*_k - \frac{f_k}{L_v}\beta^*_{vi}\big)\Big]\Big]$.*
2. *For each campaign k, we have $\alpha^*_k \in [0, c_k]$. Furthermore, either $\alpha^*_k = c_k$, or the demand constraint binds with no under-delivery, i.e., $\sum_{(v,i)\in\Gamma(k)} s_{vi} x^*_{vik} = r_k$. The optimal solution never over-delivers a campaign.*
3. *For each supply node (v,i), we have $\beta^*_{vi} \in \Big[0,\ \max_{k\in\Gamma(v,i)} \frac{w_k+\alpha^*_k}{f_k} L_v\Big]$. Furthermore, either $\beta^*_{vi} = 0$ or the supply constraint binds, i.e., $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x^*_{vik} = \delta_{vi}$.*
4. *The optimal solution to (RA-δ) is unique.*

Proof. We use the Karush-Kuhn-Tucker conditions to derive the results. Without loss of generality, we assume $\delta_{vi} > 0$ for all supply nodes (v,i); if $\delta_{vi} = 0$ we simply delete supply node (v,i), which would have an effective supply of 0, as a preprocessing step. The full Lagrangian of (RA-δ) is given

by:

$$\begin{aligned}\mathcal{L}(x,u;\alpha,\beta,\gamma,\varphi) &= \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}w_k}{2\theta_k}\left(x_{vik}-\theta_k\right)^2 \;+\; \sum_k c_k u_k \;+\; \sum_k \alpha_k\Big(r_k - \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik} - u_k\Big) \\ &\quad + \sum_{v,i}\beta_{vi}s_{vi}\Big(\sum_{k\in\Gamma(v,i)}\frac{f_k}{L_v}x_{vik} - \delta_{vi}\Big) + \sum_{v,i}\sum_{k\in\Gamma(v,i)}\left((\gamma^U_{vik}-\gamma^L_{vik})x_{vik} - \gamma^U_{vik}\right) \;-\sum_k \varphi_k u_k \\ &= \sum_{v,i}\sum_{k\in\Gamma(v,i)}\left(\frac{s_{vi}w_k}{2\theta_k}\left(x_{vik}-\theta_k\right)^2 \;-\; \Big(s_{vi}\alpha_k - \frac{f_k}{L_v}s_{vi}\beta_{vi} + \gamma^L_{vik} - \gamma^U_{vik}\Big)x_{vik} \;-\; \gamma^U_{vik}\right) \\ &\quad + \sum_k\Big((c_k-\alpha_k-\varphi_k)u_k + r_k\alpha_k\Big) \;-\; \sum_{v,i} s_{vi}\delta_{vi}\beta_{vi}.\end{aligned}$$

Dual Feasiblity:

- $\alpha_k,\ \beta_{vi},\ \gamma^U_{vik},\ \gamma^L_{vik},\ \varphi_k\ \geq 0.$

Stationarity:

- (ST1): $\frac{\partial\mathcal{L}}{\partial x_{vik}} = \frac{s_{vi}w_k}{\theta_k}\left(x_{vik}-\theta_k\right) + s_{vi}\frac{f_k}{L_v}\beta_{vi} - s_{vi}\alpha_k + \gamma^U_{vik} - \gamma^L_{vik} = 0$
  $\quad\rightarrow\quad x^*_{vik} = \theta_k + \frac{\theta_k}{w_k}\Big(\alpha^*_k - \frac{f_k}{L_v}\beta^*_{vi} + \frac{\gamma^{L*}_{vik}-\gamma^{U*}_{vik}}{s_{vi}}\Big).$
- (ST2): $\frac{\partial\mathcal{L}}{\partial u_k} = c_k - \alpha_k - \varphi_k = 0 \quad\rightarrow\quad \alpha^*_k = c_k - \varphi^*_k.$

Complementary Slackness:

- (CS1): Either $\gamma^{U*}_{vik} = 0$ or $x^*_{vik} = 1$, and either $\gamma^{L*}_{vik} = 0$ or $x^*_{vik} = 0$.
- (CS2): Either $\varphi^*_k = 0$ or $u^*_k = 0$.
- (CS3): Either $\alpha^*_k = 0$ or the demand constraint is binding: $\sum_{(v,i)\in\Gamma(k)} s_{vi}x^*_{vik} + u^*_k = r_k$.
- (CS4): Either $\beta^*_{vi} = 0$ or the supply constraint is binding, i.e., $\sum_{k\in\Gamma(v,i)}\frac{f_k}{L_v}x^*_{vik} = \delta_{vi}$.

Proof of Part 1. Conditions (ST1) and (CS1) together imply that $x^*_{vik} = \theta_k + \frac{\theta_k}{w_k}\big(\alpha^*_k - \frac{f_k}{L_v}\beta^*_{vi}\big)$ whenever this quantity falls within $(0,1)$, because the variable x^*_{vik} is not at its lower or upper bound and $\gamma^{L*}_{vik} = \gamma^{U*}_{vik} = 0$. If this quantity is negative, then $\gamma^{U*}_{vik} = 0$ and γ^{L*}_{vik} will be just high enough to make $x^*_{vik} = 0$. Similarly, if this quantity is greater than 1, then $\gamma^{L*}_{vik} = 0$ and γ^{U*}_{vik} will be just high enough to reduce its value to exactly 1. Therefore: $x^*_{vik} \equiv g_{vik}(\alpha^*_k,\beta^*_{vi}) = \min\Big[1,\max\big[0,\theta_k + \frac{\theta_k}{w_k}\big(\alpha^*_k - \frac{f_k}{L_v}\beta^*_{vi}\big)\big]\Big] \equiv \mathrm{sat}\Big[0,1,\theta_k + \frac{\theta_k}{w_k}\big(\alpha^*_k - \frac{f_k}{L_v}\beta^*_{vi}\big)\Big]$. The "sat" function notation is common in optimal control theory.

Proof of Part 2. Condition (ST2) together with dual feasibility implies that $\alpha^*_k \in [0,c_k]$. Under-delivery can only occur when $u_k > 0$ which by (CS2) requires $\varphi^*_k = 0$, which from (ST2) implies $\alpha^*_k = c_k$. If $0 < \alpha^*_k < c_k$, then $\varphi^*_k > 0$ per (ST2), and $u^*_k = 0$ per (CS2), and from (CS3) we can conclude that the demand constraint is binding with no under-delivery: $\sum_{(v,i)\in\Gamma(k)} s_{vi}x^*_{vik} = r_k$. For the case of $\alpha^*_k = 0$, we know from (CS2) that $u^*_k = 0$ but (CS3) implies $\sum_{(v,i)\in\Gamma(k)} s_{vi}x^*_{vik} \geq r_k$

which suggests that the demand constraint may not be binding. However we can show that over-delivery will never occur and the constraint is in fact binding at $\alpha_k^* = 0$. For that, we establish also that $\sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}^* \leq r_k$ when $\alpha_k^* = 0$:

$$\begin{aligned}
\sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}^* &= \sum_{(v,i)\in\Gamma(k)} s_{vi}g_{vik}(0,\beta_{vi}^*) \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\min\left[1,\max\left[0,\theta_k\left(1-\frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right)\right]\right] \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\max\left[0,\theta_k\left(1-\frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right)\right] \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k\max\left[0,\ 1-\frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right] \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k \quad = \quad r_k. \qquad (15)
\end{aligned}$$

The first inequality follows from the definition of $\min[\cdot]$, and the second inequality is due to the fact that $\max\left[0,\ 1-\frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right]$ is a quantity between 0 and 1. The last equality is due to the definition of $\theta_k = r_k/\sum_{(v,i)\in\Gamma(k)} s_{vi}$. Note that in case of truncation $\theta_k = \min\left[1, r_k/\sum_{(v,i)\in\Gamma(k)} s_{vi}\right]$, we still have $\sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k \leq r_k$ which is the desired result.

Proof of Part 3. It is clear that $x_{vik}^* = g_{vik}(\alpha_k^*,\beta_{vi}^*) = 0$ if $\beta_{vi}^* \geq \frac{w_k+\alpha_k^*}{f_k}L_v$. Therefore, if $\beta_{vi}^* \geq \max_{k\in\Gamma(v,i)}\frac{w_k+\alpha_k^*}{f_k}L_v$ (a strictly positive quantity), then $\sum_{k\in\Gamma(v,i)}\frac{f_k}{L_v}x_{vik}^* = 0 < \delta_{vi}$, which implies that the supply constraint does not bind and a strictly positive β_{vi}^* value is invalid. Therefore, it should always be that $\beta_{vi}^* \leq \max_{k\in\Gamma(v,i)}\frac{w_k+\alpha_k^*}{f_k}L_v$. The second statement in part 3 is due to condition (CS4).

Proof of Part 4. We showed in part 2 of the theorem that over-delivery never occurs. Therefore, we can eliminate u_k variables from (RA-δ) by replacing $u_k = r_k - \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}$.

$$\begin{aligned}
\text{(RA-}\delta\text{)} \quad \equiv \quad \text{Minimize} \quad & \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}}{2\theta_k} w_k\left(x_{vik}-\theta_k\right)^2 + \sum_k c_k\Big(r_k - \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}\Big) && \text{(16a)} \\
\text{s.t.} \quad & \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik} \leq r_k \qquad \forall k && \text{(16b)} \\
& \sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v}x_{vik} \leq \delta_{vi} \qquad \forall v,i && \text{(16c)} \\
& 0 \leq x_{vik} \leq 1 \qquad \forall v,i,k\in\Gamma(v,i) && \text{(16d)}
\end{aligned}$$

The constraint (16b) corresponds to $u_k \geq 0$. It is easy in this form to see that the objective function is strictly convex: The Hessian matrix is diagonal with elements $s_{vi}w_k/\theta_k > 0$ which make it strictly positive definite. The constraints are linear and therefore define a convex feasible set. A strictly convex function has a unique global minimum over a convex set. □

G Proof of Theorem 2 (Convergence and Optimality of Modified SHALE)

Theorem. *Given a vector of impression utilization factors δ, the Modified SHALE Algorithm converges to the optimal dual solution for (RA-δ) as long as either (i) all α_k values are initialized to zero, or (ii) we initialize $\alpha_k = \alpha'_k, \forall k \in \mathcal{K}$ where α' is the optimal dual solution to (RA-δ') for which $\delta' \geq \delta$ componentwise.*

Proof. We present the proof in two parts. First, we prove that the algorithm converges by showing that, when initialized properly, the α_k values strictly increase following each Step-2 update (unless the value is maxed-out at c_k). Since each α_k is bounded above by c_k, the algorithm must converge. Second, we prove optimality by showing that the resulting solution satisfies all KKT conditions. Since the problem (RA-δ) is convex, any solution that satisfies all KKT conditions must be optimal. Following the convergence and optimality proof, we also discuss the optimality gap when the algorithm is terminated early before full convergence.

Convergence:

Let α_k^t and β_{vi}^t denote the dual values computed in iteration t of SHALE, and let $r_k(\alpha_k, \boldsymbol{\beta}) = \sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik} = \sum_{(v,i)\in\Gamma(k)} s_{vi}g_{vik}(\alpha_k, \beta_{vi})$ denote the volume of satisfied demand (reach) for campaign k given the current dual vectors $\boldsymbol{\alpha}^t$ and $\boldsymbol{\beta}^t$ in iteration t. Therefore, $r_k(\alpha_k^{t-1}, \boldsymbol{\beta}^t)$ gives the satisfied demand following the $\boldsymbol{\beta}$ updates in Step-1 of iteration t, and $r_k(\alpha_k^t, \boldsymbol{\beta}^t)$ shows this quantity following the $\boldsymbol{\alpha}$ updates in Step-2. We have:

$$\begin{aligned}
\left|r_k(\alpha_k^t, \boldsymbol{\beta}^t) - r_k(\alpha_k^{t-1}, \boldsymbol{\beta}^t)\right| &= \left|\sum_{(v,i)\in\Gamma(k)} s_{vi}g_{vik}(\alpha_k^t, \beta_{vi}^t) - s_{vi}g_{vik}(\alpha_k^t, \beta_{vi}^t)\right| \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|g_{vik}(\alpha_k^t, \beta_{vi}^t) - g_{vik}(\alpha_k^t, \beta_{vi}^t)\right| \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|\operatorname{sat}\Big[0,1,\theta_k + \frac{\theta_k}{w_k}\Big(\alpha_k^t - \frac{f_k}{L_v}\beta_{vi}^t\Big)\Big] - \operatorname{sat}\Big[0,1,\theta_k + \frac{\theta_k}{w_k}\Big(\alpha_k^{t-1} - \frac{f_k}{L_v}\beta_{vi}^t\Big)\Big]\right| \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|\frac{\theta_k}{w_k}\Big(\alpha_k^t - \alpha_k^{t-1}\Big)\right| \\
&= \frac{r_k}{w_k}\left|\alpha_k^t - \alpha_k^{t-1}\right| \qquad (17)
\end{aligned}$$

where the first inequality is due to the triangle inequality, and the second inequality follows from the fact that for any two numbers a and b, $\left|\min\left[1, \max[0, a]\right] - \min\left[1, \max[0, b]\right]\right| \leq |a - b|$. (Equality occurs when both a and b are within $[0, 1]$, and in all other cases the length of interval $[a, b]$ is being truncated by the $\min[1, \max[0, \cdot]]$ operation, either from above (at 1) or below (at 0), or both). The last equality follows from the definition of $\theta_k = r_k / \sum_{(v,i)\in\Gamma(k)} s_{vi}$.

Condition 1 *(Sufficient Condition for Convergence):* There exists an iteration t_0, such that

following the Step-1 ($\boldsymbol{\beta}$ updates) we observe $r_k(\alpha_k^{t_0-1}, \boldsymbol{\beta}^{t_0}) \leq r_k$ for all $k \in \mathcal{K}$. That is, no campaign is over-delivered.

In the Step-2 ($\boldsymbol{\alpha}$ updates) we either set $\alpha_k^t = c_k$ (the value of α_k is maxed-out and campaign k will face under-delivery), or whenever possible, we set α_k^t such that $r_k(\alpha_k^t, \boldsymbol{\beta}^t) = r_k$. In the latter case, if Condition 1 holds at iteration t_0, then (17) suggests:

$$r_k(\alpha_k^t, \boldsymbol{\beta}^t) - r_k(\alpha_k^{t-1}, \boldsymbol{\beta}^t) \;=\; r_k - r_k(\alpha_k^{t-1}, \boldsymbol{\beta}^t) \;\leq\; \frac{r_k}{w_k}(\alpha_k^{t_0} - \alpha_k^{t_0-1})$$

$$\Rightarrow \qquad \alpha_k^{t_0} \;\geq\; \alpha_k^{t_0-1} + w_k\Big(1 - \frac{r_k(\alpha_k^{t_0-1}, \boldsymbol{\beta}^{t_0})}{r_k}\Big) \;\geq\; \alpha_k^{t_0-1} \tag{18}$$

That is, no α_k value will decrease in the Step-2 update, when Condition 1 holds. Note that every $g_{vik}(\cdot)$ term and therefore $r_k(\cdot)$ is non-decreasing in α_k. Therefore, $\alpha_k^t \geq \alpha_k^{t-1}$ implies $r_k(\alpha_k^t, \boldsymbol{\beta}^t) \geq r_k(\alpha_k^{t-1}, \boldsymbol{\beta}^t)$ and vice versa. Hence, we can remove the absolute values from both sides of (17) when $r_k(\alpha_k^t, \boldsymbol{\beta}^t) = r_k \geq r_k(\alpha_k^{t-1}, \boldsymbol{\beta}^t)$ which is assumed to hold by Condition 1.

We now show that following the $\boldsymbol{\beta}$ update in Step-1 of iteration t_0+1, Condition 1 will hold for iteration t_0+1 as well, proving that α_k values will again strictly increase or max-out at c_k in t_0+1 and all subsequent iterations. Note that every $g_{vik}(\cdot)$ term and therefore $r_k(\cdot)$ is non-increasing in $\boldsymbol{\beta}$. At the beginning of Step-1 of iteration t_0+1 one of the following could happen for each supply node (v,i):

1. The supply constraint is binding: $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0}) = \delta_{vi}$. This happens if no α_k from campaigns $k \in \Gamma(v,i)$ that target (v,i) has been changed in the past iteration. In this case, no update to β_{vi} value is necessary: $\beta_{vi}^{t_0+1} = \beta_{vi}^t \geq 0$.

2. The supply constraint is non-binding and not violated: $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0}) < \delta_{vi}$. We know from (18) that all $\alpha_k^{t_0} \geq \alpha_k^{t_0-1}$ and that $g_{vik}(\cdot)$ is non-decreasing in α_k. Therefore, it must have been that $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^{t_0-1}, \beta_{vi}^{t_0}) < \delta_{vi}$, i.e., the supply constraint was not binding following the Step-1 update of iteration t_0 and $\beta_{vi}^{t_0} = 0$. To make the supply constraint bind, we need to decrease the β_{vi} value even further, which is not possible since negative values are not allowed for β_{vi}. Therefore, the β_{vi} value remains at zero with no change: $\beta_{vi}^{t_0+1} = \beta_{vi}^{t_0} = 0$, and the supply constraint remains non-binding.

3. The supply constraint is violated: $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0}) > \delta_{vi}$. This is the most likely situation for any supply constraint that was binding after Step-1 in iteration t_0. In this case, we can always increase β_{vi} as much as necessary to decrease the left-hand side until $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0+1}) = \delta_{vi}$. In this case we will have $\beta_{vi}^{t_0+1} > \beta_{vi}^{t_0}$. We should point out that the upper-bound for β_{vi} suggested in Part 3 of Theorem 1 is the threshold beyond which the left-hand side of the supply constraint (v,i) becomes zero, which ensures feasibility for any $\delta_{vi} > 0$. Therefore, it is not restrictive and is only deduced to eliminate uninfluential β_{vi} values from the search space.

Overall, we observe that no β_{vi} value will decrease in the Step-1 update. Therefore:

$$r_k(\alpha_k^{t_0}, \boldsymbol{\beta}^{t_0+1}) \;=\; \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0+1}) \;\le\; \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0}) \;=\; r_k(\alpha_k^{t_0}, \boldsymbol{\beta}^{t_0}) \;\le\; r_k \tag{19}$$

which is the Condition 1 for Iteration t_0+1. This implies that all $\alpha_k^{t_0+1} \ge \alpha_k^{t_0}$ in Step-2 of iteration t_0+1, per (18), and therefore all $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ values will monotonically increase in all iterations $t \ge t_0$, and Condition 1 will be maintained throughout. Since α_k is bounded above by c_k, the algorithm must converge.

In summary, Condition 1 requires that no campaign is over-delivered. Then in each α_k update, we seek to eliminate under-delivery for each campaign k by increasing α_k as much as possible (and α_k maxed-out at c_k implies we could not fully eliminate under-delivery and $u_k > 0$). As a result of increasing α_k value, we increase x_{vik} for all $(v,i) \in \Gamma(k)$ which may consequently violate the supply constraint for some of those viewer types. In the subsequent β_{vi} update, we increase β_{vi} (decrease x_{vik} for all $k \in \Gamma(v,i)$) to recover supply feasibility at those nodes. If the supply constraint has leftover excess and $\beta_{vi} > 0$ (obviously violating complementary slackness), instead, we decrease β_{vi} (increase x_{vik} for all $k \in \Gamma(v,i)$) as much as possible (considering non-negativity) and try to allocate as much supply as available. We showed that once Condition 1 holds, and at least one round of $\boldsymbol{\beta}$ updates has been performed to correct complementary slackness, then we never need to decrease β_{vi} values as they will continue to take their lower-bound of 0 when the corresponding supply constraint non-binding.

Initialization (Satisfying Condition 1):

Now we show that with proper initialization of α_k values, we can make Condition 1 hold from the first iteration. This is trivial when all $\alpha_k^0 = 0$. The maximum $r_k(\alpha_k^0, \boldsymbol{\beta}^1)$ is attained when all $\beta_{vi}^1 = 0$, therefore $r_k(\alpha_k^0, \boldsymbol{\beta}^1) \le \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(0,0) = \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k = r_k$. The original proof of convergence for the SHALE algorithm, provided in Bharadwaj et al. (2012), only explores the initialization of $\alpha_k^0 = 0$, which is assuming the worst case values for β_{vi}, i.e., when they are all set to zero.

In our framework, we claim that to solve (RA-δ) following an adjustment (reduction) in δ_{vi} values, we can initialize our modified SHALE algorithm using the current optimal α values prior to adjustment. To see this, assume that the current optimal dual solution to (RA-δ') is $\alpha_k^*(\delta')$ and $\beta_{vi}^*(\delta')$. Clearly, $r_k(\alpha_k^*(\delta'), \boldsymbol{\beta}^*(\delta')) \le r_k$ (see (15) in Appendix F that shows over-delivery never occurs in the optimal solution). Assume we need to solve a new instance (RA-δ) in which $\delta_{vi} \le \delta'_{vi}$ for all (v,i). Initializing $\alpha_k^0 = \alpha_k^*(\delta')$, note that if at any node (v,i) we happen to have $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^0, \beta_{vi}^*(\delta')) \le \delta_{vi} \le \delta'_{vi}$, then we naturally obtain $\beta_{vi}^1 = \beta_{vi}^*(\delta')$. In the case of $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^0, \beta_{vi}^*(\delta')) > \delta_{vi}$ we need to increase the β_{vi} value to decrease the left-hand side until the constraint binds: $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^0, \beta_{vi}^1) = \delta_{vi}$. In this case, we have $\beta_{vi}^1 > \beta_{vi}^*(\delta)$. Overall, we can conclude that $\beta_{vi}^1 \ge \beta_{vi}^*(\delta)$ for every (v,i). From (19) we obtain that $r_k(\alpha_k^0, \beta^1) \le r_k(\alpha_k^*(\delta), \beta^*(\delta)) \le r_k$ which meets Condition 1 for iteration $t_0 = 1$.

Optimality:

We now show that the solution obtained from Modified SHALE satisfies all KKT conditions for the problem (RA-δ). Since (RA-δ) is a convex problem, the solution must be optimal.

Dual feasibility is always maintained by limiting the search space for α_k and β_{vi} to non-negative values. The stationarity condition (ST1) for variable x_{vik} together with complementary slackness conditions (CS1) for the basic bounds $0 \leq x_{vik} \leq 1$ are also maintained in every step by the virtue of setting $x_{vik} = g_{vik}(\alpha_k, \beta_{vi})$. The stationarity condition (ST2) for slack variables u_k, and the complementary slackness conditions (CS2) for $u_k \geq 0$ and (CS3) for the demand constraint of campaign k are all achieved following the α_k update in Step-1 of the algorithm. The complementary slackness condition (CS4) for the supply constraint for the viewer type (v, i) is achieved following the β_{vi} updates in Step-2 of the algorithm.

As a part of proving the convergence of the algorithm, we showed that no campaign will experience over-delivery in any iteration subsequent to meeting Condition 1. We also showed that the primal solution always satisfies the supply constraints after the Step-1 $\boldsymbol{\beta}$ updates. So, after the $\boldsymbol{\alpha}$ values converge, the final adjustment of $\boldsymbol{\beta}$'s will ensure complete primal feasibility, dual feasibility, complementary slackness, and stationarity. □

Performance Gap:

The optimality bound, due to Bharadwaj et al. (2012), is based on the argument that for any $t \geq t_0$, if for some k with $\alpha_k^t \neq c_k$ we have $r_k(\alpha_k^{t-1}, \beta^t) \leq (1-\varepsilon) r_k$, then (18) implies $\alpha_k^t \geq \alpha_k^{t_0-1} + w_k\varepsilon$. That is, α_k increases by at least $w_k\varepsilon$. If $\alpha_k^0 = 0$, then at most $c_k/(w_k\varepsilon)$ of such adjustments will be made on α_k. This suggests that after a worst-case scenario of $t \geq |\mathcal{K}| \cdot \max_k\{c_k/(w_k\varepsilon)\}$ iterations, all campaigns for which α_k is not maxed-out at c_k (i.e., are chosen be delivered fully in the optimal solution) should be delivered within an ϵ-fraction of their r_k.

H Geometric Illustration of δ Updates

In the essence, our δ updates during the feasibility phase of Pattern-HCG try to ensure the feasibility of all user-supply constraints (4c) in (PA) by appropriately adjusting the impression-supply constraints (3c) in the aggregate planning problem (RA-δ). In this section we provide a geometric comparison of these two types of constraints, show how we can easily calculate a lower- and upper-bound for each δ_{vi}, and point out the possibility of more advanced updating rules than (7) which could improve the performance of Pattern-HCG.

In solving (PA) we take the approach of relaxing the user-supply constraint (4c) so a feasible solution is guaranteed and easy to construct to initialize our column generation procedure. However, note that the constraint set (4b) together with the impression-supply constraints (3c) from (RA-δ),

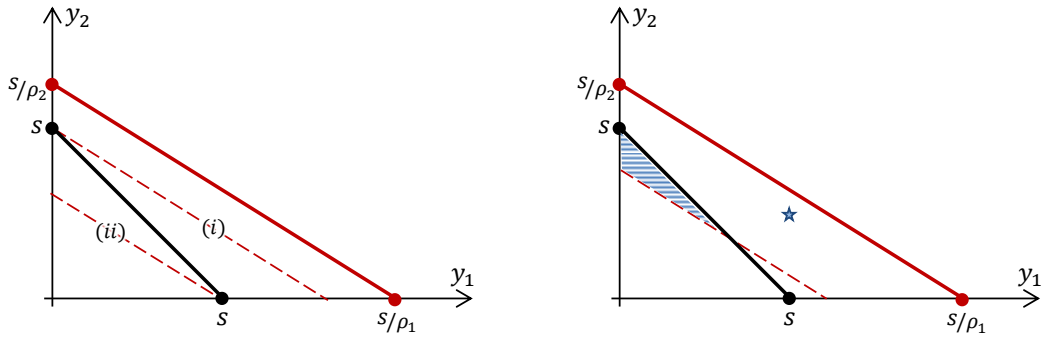
imply:

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} = \sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v}\Big(\frac{1}{s_{vi}} \sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip}\Big) = \frac{1}{s_{vi}} \sum_{p\in\mathcal{P}_{vi}} \Big(\frac{\sum_{k\in\Gamma(v,i)} f_k b_{kp}}{L_v}\Big) y_{vip}$$
$$= \frac{1}{s_{vi}} \sum_{p\in\mathcal{P}_{vi}} \rho_{vip} y_{vip} \;\le\; \delta_{vi}$$

where $\rho_{vip} = \sum_{k\in\Gamma(v,i)} f_k b_{kp}/L_v$ is the utilization ratio of pattern $p \in \mathcal{P}_{vi}$ and is less than one (as per (5b)). Figure 9 illustrates the implied constraint $\sum_{p\in\mathcal{P}_{vi}} \rho_{vip} y_{vip} \le s_{vi}\delta_{vi}$ (red lines), against the original symmetric constraint $\sum_{p\in\mathcal{P}_{vi}} y_{vip} \le s_{vi}$ (solid black line), for a particular supply node with two possible patterns (we suppress the (v,i) subscripts for readability).

Let $\delta_{vi}^{\min}$ and $\delta_{vi}^{\max}$ respectively denote the minimum (non-empty) and maximum impression utilization rates possible for supply node (v,i). Obviously, $\delta_{vi}^{\min} = \min_{k\in\Gamma(v,i)}\{f_k\}/L_v$, i.e., the pattern consisting of only the campaign with smallest f_k; and $\delta_{vi}^{\max}$ can be determined by solving a binary knapsack problem $\max_{b_k\in\{0,1\}}\{\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} b_k : \sum_{k\in\Gamma(v,i)} f_k b_k \le L_v\}$ which finds the best packing of campaigns $k \in \Gamma(v,i)$ possible over L_v slots. The parameter δ_{vi} which shows the achieved average level of impression utilization in node (v,i) should therefore fall within the range $[\delta_{vi}^{\min}, \delta_{vi}^{\max}]$. The two red dashed lines on Figure 9(a) illustrate the implied constraint $\sum_{p\in\mathcal{P}_{vi}} \rho_{vip} y_{vip} \le s_{vi}\delta_{vi}$ when δ_{vi} is exactly at $\delta_{vi}^{\min}$ or $\delta_{vi}^{\max}$.

In the absence of the user-supply constraint (4c), i.e., the solid black line, our approach is to adjust the δ_{vi} values until the implied constraints $\sum_{p\in\mathcal{P}_{vi}} \rho_{vip} y_{vip} \le s_{vi}\delta_{vi}$ push the optimal solution of (PA) to satisfy $\sum_{p\in\mathcal{P}_{vi}} y_{vip} \le s_{vi}$. Considering the slope differences between these two types of constraints, Figure 9(b) shows that a certain portion of the feasible region (hatched in blue) may be cut off. This may cause the solution produced by Pattern-HCG to be suboptimal with respect to the primary aggregate quality objective. The degree of this suboptimality depends on the relative values of ρ_{vip} across all nodes and cannot be characterized in closed form. Setting $\delta_{vi} = \delta_{vi}^{\min}$ at all nodes of (RA-δ) causes all (PA) problems to be feasible (i.e., the δ-adjusted impression supply constraints dominate all user supply constraints) and the resulting solution provides the worst-case suboptimality of our approach. To numerically assess this optimality gap, we solved some instances using both Pattern-HCG as well as the monolithic formulation presented in Appendix E. Since we were interested in assessing the optimality gap of the primary aggregate quality objective, we ignored disaggregate pattern quality by setting $\pi(\mathbf{b}) = 0$. Note that generally speaking the monolithic formulation, which has a composite objective that sums together aggregate and disaggregate pattern quality terms, solves a different problem than our R&F planning problem which has both a primary aggregate quality objective and a subordinate disaggregate quality objective. However, when $\pi(\mathbf{b}) = 0$ the monolithic formulation directly maximizes aggregate pattern quality, and thus can be used to find a solution to our R&F ad planning problem that is optimal for the primary aggregate pattern quality objective. The monolithic formulation solves the reach allocation and pattern assignment components simultaneously, whereas Pattern-HCG solves them sequentially coupled with δ-updates which leads to sub-optimal solutions. However,



(a) Dashed lines (i) and (ii) illustrate the translated impression supply constraint if δ is set to $\delta^{\max}$ and $\delta^{\min}$, respectively. It is never beneficial to set δ outside this range.

(b) Current optimal solution (star symbol), the implied constraint following the δ update (dashed red line), and the area which is cut off from the feasible region (hatched).

Figure 9: Geometric illustration of user supply constraint (solid black line) vs. the translated impression supply constraint adjusted by δ (red lines). The solid red line illustrates the case of $\delta = 1$.

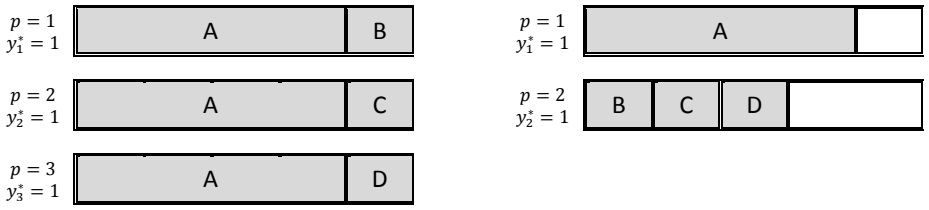
our numerical tests on realistic yet smaller instances that match our industry data suggest that the solution produced by Pattern-HCG is only 1-3 percent suboptimal with respect to the primary aggregate quality objective. Overall, we feel this is reasonable given the many advantages that our hierarchical formulation has over the monolithic formulation, as described in §5 and Appendix E.

Moreover, we note that in §5.4 we adopted the simplest update rule for δ values, and that more advanced update rules may tighten the optimality gap. For instance, we noticed if we update only a fraction (and not all) of the δ_{vi} values at each iteration (especially, if chosen based on the smallest β_{vi} value, i.e., to have the least impact on the objective of (RA-δ)), the optimality gap can be further reduced.

I Equivalence of the Scrap-minimizing and Roll-minimizing Cutting-stock Problems

In this section we show that when over-production is not allowed, i.e., demand constraints are expressed as equality, the cutting stock (pattern assignment) problem that minimizes scrap (excess) is equivalent to one that minimizes the number of stock rolls (individual users) used. We used this property in §5.4 to argue that our update rule for δ values is conservative.

Consider the classic cutting stock problem where a manufacturer has an infinite stock of metal rolls (or rods) of fixed length L, and there is a demand r_k for pieces of length $f_k < L$. The manufacturer may minimize scrap (pieces of roll that are not of usable length and must be scrapped) by generating a number of cutting patterns, and determining the number of times to use (i.e., cut stock from) each pattern. Using a_{kp} to denote the number of times piece k (of length f_k) is cut from a roll when pattern p is used, $\pi_p = L - \sum_k a_{kp} f_k$ to denote the amount of scrap produced from each roll cut using pattern p, and variables y_p to denote how many rolls are cut using pattern



(a) Waste-Minimizing Solution (b) Roll-minimizing Solution

Figure 10: Comparison of optimal solutions to a cutting stock problem when demand constraints are expressed as inequalities (i.e., over-production is allowed)

p, the pattern assignment math program is: $\min\left\{\sum_p \pi_p y_p \mid \sum_p a_{kp} y_p \geq r_k,\ y_p \geq 0\right\}$.

Substituting the definition of π_p into the objective function, we get:

$$\sum_p \pi_p y_p = \sum_p \left(L - \sum_k a_{kp} f_k\right) y_p = \sum_p L y_p - \sum_p \left(\sum_k f_k a_{kp}\right) y_p$$
$$\equiv L\left(\sum_p y_p\right) - \sum_k f_k \left(\sum_p a_{kp} y_p - r_k\right) \qquad \text{(differs only by a constant, } \textstyle\sum_k f_k r_k\text{)}$$

Therefore, if the demand constraints are expressed as equality constraints and do not allow for over-production (as is the case in our Pattern Assignment problem), the scrap-minimizing objective $\sum_p \pi_p y_p$ is *equivalent* to the objective that minimizes the number of raw rolls $\sum_p y_p$ (in our case, the number of unique users) used, and vice versa.

However, when the demand constraints are written in inequality form (allowing demand to be exceeded) the scrap-minimizing problem, as written above, may use more raw rolls to improve the packing at the expense of over-producing some of the final goods. For example, consider four products of lengths $f_A = 4$, $f_B = f_C = f_D = 1$ that each have a single unit of demand $r_k = 1$. With raw rolls of length $L = 5$, Figure 10 shows that the scrap-minimizing solution may use each of the following three patterns $\{AB, AC, AD\}$ once. Three rolls are used to achieve zero scrap, but 2 units of product A are produced in excess of the amount demanded. In contrast, the roll-minimizing solution may use each of the following two patterns $\{A, BCD\}$ once, scrapping 3 units of raw material, but only 2 rolls are used rather than 3 (Note that neither problem has a unique solution; the solutions illustrated here are among the possible optimal solutions which we may get following a column generation procedure).

Finally, we note that if the over-production of goods is undesired (e.g., cannot be sold), the scrap-minimizing objective should be defined as $\sum_p \pi_p y_p + \sum_k f_k \left(\sum_p a_{kp} y_p - r_k\right)$, which also counts over-production as scrap. With this objective, the scrap-minimizing problem is again equivalent to the roll-minimizing problem. Now, the roll-minimizing solution $\{A, BCD\}$ which scraps 3 units is cheaper than the solution $\{AB, AC, AD\}$ which over-produces product A by 2 units and thus creates 8 units of scrap.