Electronic Companion:

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

Ali Hojjat

Peter T. Paul College of Business, University of New Hampshire, ali.hojjat@unh.edu

John Turner

Paul Merage School of Business, University of California Irvine, john.turner@uci.edu

Suleyman Cetintas, Jian Yang

Yahoo Research, Sunnyvale, CA, {cetintas,jianyang}@yahoo-inc.com

This e-companion accompanies the full paper of the same title.

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

EC.1 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 $\left(b_k\right)_{k \in \Gamma(v,i)}$ 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 §EC.2.
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).

EC.2 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 §EC.4 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:

$$\pi_{pacing}(\mathbf{b}) \;=\; \text{Minimize} \quad \sum_{k\in\Gamma(v,i)} \sum_{\ell=1}^{L_v} \left(\sum_{\ell'=1}^{\ell} z_{k\ell'} - b_k T_\ell \right)^2 \tag{11a}$$

$$\sum_{\ell=1}^{L_v} z_{k\ell} \;=\; b_k f_k \qquad \forall k \in \Gamma(v,i) \tag{11b}$$

$$\sum_{k\in\Gamma(v,i)} z_{k\ell} \;\leq\; 1 \qquad \forall \ell = 1, ..., L_v \tag{11c}$$

$$z_{k\ell} \in \{0,1\} \tag{11d}$$

Constraint (11b) 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 (11c) 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 (11b)).

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 (11) in isolation but rather we wish to solve (11) as an inner-optimization within (PG). Unfortunately, we cannot transform (11) into an assignment problem when the $\mathbf{b}$ vector is also a decision variable. Instead, to integrate (11) into (PG), we use (11a) as the objective and include the constraints (11b,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 (11) 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:

$$\pi_{pacing}(\mathbf{b}) \;=\; \text{Minimize} \quad \sum_k \sum_{j_k=2}^{f_k} \left| Z_{j_k} - Z_{(j-1)_k} - \frac{L_v}{f_k} b_k \right| \tag{12a}$$

$$\sum_{j_k=1}^{f_k} \sum_{\ell=1}^{L_v} z_{j_k \ell} \;=\; f_k b_k \qquad \forall k \tag{12b}$$

$$\sum_k \sum_{j_k=1}^{f_k} z_{j_k \ell} \;\leq\; 1 \qquad \forall \ell = 1, ..., L_v \tag{12c}$$

$$Z_{j_k} \;=\; \sum_{\ell=1}^{L_v} \ell z_{j_k \ell} \qquad \forall k,\; j_k = 1, ..., f_k \tag{12d}$$

$$Z_{j_k} \;\geq\; Z_{(j-1)_k} + 1 \qquad \forall k,\; j_k = 2, ..., f_k \tag{12e}$$

$$z_{j_k \ell} \in \{0, 1\}, \quad Z_{j_k} : \text{Integers} \tag{12f}$$

Constraints (12b) and (12c) perform the same function as (11b) and (11c). Constraint (12d) establishes the relationship between variables $z_{j_k \ell}$ and Z_{j_k}, and constraint (12e) 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 (12), 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 (12) 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 (12a) as the objective and (12b-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 one should prefer to extend (PG) using (11) rather than (12) to pace ads uniformly.

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.

EC.3 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 §EC.2 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 §EC.2, 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 §EC.2.

Sub-problem (PG) can be cast as:

$$\text{Minimize} \quad \pi(\mathbf{b}) - \sum_{k\in\Gamma(v,i)} \bar{\alpha}^*_{vik} b_k \tag{13a}$$

$$\text{s.t.} \quad \sum_{k\in\Gamma(v,i)} f_k b_{kh} \le L_v \qquad \forall h = 1,...,H \tag{13b}$$

$$b_k \equiv \sum_{h=1}^{H} b_{kh} \le 1 \qquad \forall k \in \Gamma(v,i) \tag{13c}$$

$$b_{kh} \in \{0,1\}, \qquad \forall k \in \Gamma(v,i),\ \forall h = 1,...,H \tag{13d}$$

Constraint (13b) 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 (13c) 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 §EC.2 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 (13) 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).

EC.4 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:

$$\text{(PA-R):} \quad \Psi_{vi}^{(R)} := \text{Minimize} \quad \sum_{p \in \mathcal{P}_{vi}} \pi_{vip} y_{vip} \qquad \underline{\text{Duals:}} \qquad (14a)$$

$$\sum_{p \in \mathcal{P}_{vi}} \bar{b}_{kp} y_{vip} = s_{vi} x^*_{vik} \quad \forall k \in \Gamma(v, i) \qquad \bar{\alpha}_{vik}^{(R)} \text{ (free)} \qquad (14b)$$

$$\sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq s_{vi} \qquad \bar{\beta}_{vi}^{(R)} \geq 0 \qquad (14c)$$

$$y_{vip} \geq 0 \quad \forall p \in \mathcal{P}_{vi} \qquad - \qquad (14d)$$

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 (14b). The pattern generating subproblem takes the following form:

$$\text{(PG-R):} \qquad \psi_{vi}^{(R)} := \text{Maximize} \quad \sum_{k\in\Gamma(v,i)} \bar{\alpha}_{vik}^{*(R)} \underbrace{\Big(\sum_{k=1}^{\bar{L}_v} \phi_v^{(\ell)} I_{k\ell}\Big)}_{\bar{b}_k} - \pi(\mathbf{b}) \tag{15a}$$

$$\sum_{k\in\Gamma(v,i)} z_{k\ell} \le 1 \qquad \ell = 1, ..., \bar{L}_v \tag{15b}$$

$$\sum_{\ell=1}^{\bar{L}_v} z_{k\ell} = f_k b_k \qquad \forall k \in \Gamma(v,i) \tag{15c}$$

$$\sum_{\ell'=1}^{k} z_{k\ell'} \le f_k - 1 + I_{k\ell} \qquad \forall k \in \Gamma(v,i),\ \ell = 1, ..., \bar{L}_v \tag{15d}$$

$$\sum_{\ell'=1}^{k} z_{k\ell'} \ge f_k I_{k\ell} \qquad \forall k \in \Gamma(v,i),\ \ell = 1, ..., \bar{L}_v \tag{15e}$$

$$b_k, z_{k\ell}, I_{k\ell} \in \{0,1\} \tag{15f}$$

The first set of constraints (15b) ensure that at most one campaign occupies each slot. The second set of constraints (15c) 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 (15d) and (15e) are the cumulative impression counts $\bar{z}_{k\ell}$. Constraints (15d) enforce $I_{k\ell} = 1$ when $\bar{z}_{k\ell} = f_k$, whereas constraints (15e) 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 §EC.2.

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 (15a).

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.

EC.5 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}:

$$\text{Minimize} \quad \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi} w_k}{2\theta_k} (x_{vik} - \theta_k)^2 + \sum_k c_k u_k + \sum_{v,i} \sum_{p\in\mathcal{P}_{vi}} \pi_{vip} y_{vip} \tag{16a}$$

$$\text{s.t.} \quad x_{vik} = \frac{1}{s_{vi}} \sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip} \qquad \forall v,i,k \in \Gamma(v,i) \tag{16b}$$

$$\sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} + u_k \geq r_k \qquad \forall k \tag{16c}$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq 1 \qquad \forall v,i \tag{16d}$$

$$\sum_{p\in\mathcal{P}_{vi}} y_{vip} \leq s_{vi} \qquad \forall v,i \tag{16e}$$

$$0 \leq x_{vik} \leq 1 \tag{16f}$$

$$y_{vip} \geq 0,\ u_k \geq 0 \tag{16g}$$

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 (16b) 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 (16c) and (16d) are supply and demand constraints from (RA). Constraint (16e) 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 (16f) and (16g) 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 (16f) 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 (16b). The next inequality follows since each b_{kp} value is at most 1. Finally, the last inequality follows from constraint (16e).

Next, we show that the user-based supply constraint (16e) is always tighter than the impression-based supply constraint (16d). In other words, (16d) is dominated by (16e), making (16d) 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 (16b). 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 (16e).

Finally, after dropping the redundant constraints (16d) and (16f) and eliminating x_{vik} by substitution using constraint (16b), we can represent the monolithic formulation of Hojjat et al. (2014) in the following simplified form:

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

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}:

$$\gamma_{vip} \;=\; \sum_{k \in \Gamma(v,i)} \left(\frac{w_k}{\theta_k s_{vi}} \sum_{p' \in \mathcal{P}_{vi}} b_{kp'} y_{vip'} - w_k - \alpha_k \right) b_{kp} + \pi_{vip} + \beta_{vi}. \tag{17}$$

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 (17) 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 (16b), the reduced cost function (17) 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 §EC.2.

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 §EC.2). 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.

EC.6 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 $\alpha_k^* = 0$:

$$\begin{aligned}
\sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik}^* &= \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(0, \beta_{vi}^*) \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi} \min\left[1, \max\left[0, \theta_k\left(1 - \frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right)\right]\right] \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi} \max\left[0, \theta_k\left(1 - \frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right)\right] \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k \max\left[0,\ 1 - \frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right] \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k \quad = \quad r_k. \qquad (18)
\end{aligned}$$

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 $u_k = r_k - \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik}$.

$$\begin{aligned}
\text{(RA-}\delta\text{)} \quad \equiv \quad \text{Minimize} \quad & \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}}{2\theta_k} w_k \left(x_{vik} - \theta_k\right)^2 + \sum_k c_k\Big(r_k - \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik}\Big) && \text{(19a)} \\
\text{s.t.} \quad & \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} \leq r_k \qquad \forall k && \text{(19b)} \\
& \sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq \delta_{vi} \qquad \forall v, i && \text{(19c)} \\
& 0 \leq x_{vik} \leq 1 \qquad \forall v, i, k \in \Gamma(v,i) && \text{(19d)}
\end{aligned}$$

The constraint (19b) 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. □

EC.7 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:

$$
\begin{aligned}
\left|r_k(\alpha_k^t, \boldsymbol{\beta}^t) - r_k(\alpha_k^{t-1}, \boldsymbol{\beta}^t)\right| &= \left|\sum_{(v,i)\in\Gamma(k)} s_{vi}g_{vik}(\alpha_k^t, \beta_{vi}^t) - s_{vi}g_{vik}(\alpha_k^t, \beta_{vi}^t)\right| \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|g_{vik}(\alpha_k^t, \beta_{vi}^t) - g_{vik}(\alpha_k^t, \beta_{vi}^t)\right| \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|\operatorname{sat}\left[0,1,\theta_k + \frac{\theta_k}{w_k}\left(\alpha_k^t - \frac{f_k}{L_v}\beta_{vi}^t\right)\right] - \operatorname{sat}\left[0,1,\theta_k + \frac{\theta_k}{w_k}\left(\alpha_k^{t-1} - \frac{f_k}{L_v}\beta_{vi}^t\right)\right]\right| \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|\frac{\theta_k}{w_k}\left(\alpha_k^t - \alpha_k^{t-1}\right)\right| \\
&= \frac{r_k}{w_k}\left|\alpha_k^t - \alpha_k^{t-1}\right| \qquad (20)
\end{aligned}
$$

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 (20) suggests:

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

$$\Rightarrow \qquad \alpha_k^{t_0} \;\geq\; \alpha_k^{t_0-1} + w_k\Big(1 - \frac{r_k(\alpha_k^{t_0-1}, \boldsymbol{\beta}^{t_0})}{r_k}\Big) \;\geq\; \alpha_k^{t_0-1} \tag{21}$$

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 (20) 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 (21) 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:

$$r_k(\alpha_k^{t_0}, \boldsymbol{\beta}^{t_0+1}) \;=\; \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0+1}) \;\le\; \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0}) \;=\; r_k(\alpha_k^{t_0}, \boldsymbol{\beta}^{t_0}) \;\le\; r_k \tag{22}$$

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 (21), 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 (18) in §EC.6 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 (22) 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 (21) 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.

EC.8 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 (9) 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),

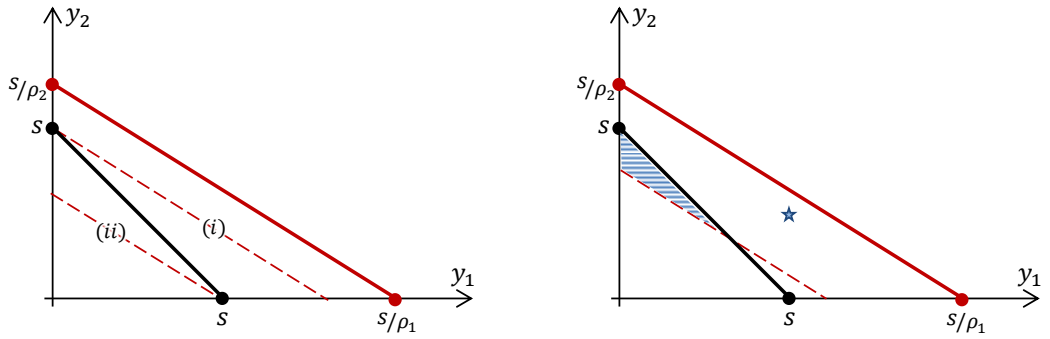
imply:

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

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 1 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 1(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 1(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 §EC.5. 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, our numerical tests on realistic yet smaller instances



(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 1: 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 $\delta = 1$.

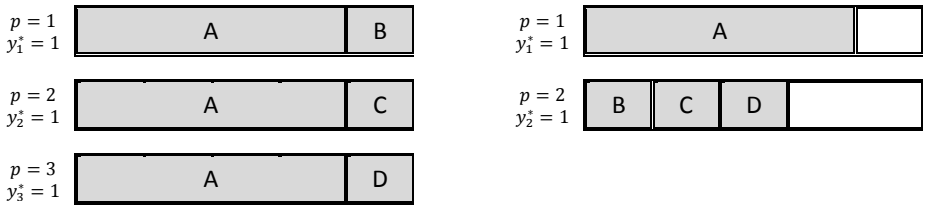
that match our industry data suggest that the solution produced by Pattern-HCG is only 1-3 percent suboptimal with respect to the primary aggregate quality objective. Overall, we feel this is reasonable given the many advantages that our hierarchical formulation has over the monolithic formulation, as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

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.

EC.9 Equivalence of 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 2: 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, $\sum_p a_{kp} y_p = r_k$, 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 2 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.

References

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*.

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.

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.

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.

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