Title
New Models and Mechanisms for the Planning and Allocation of Online Advertising

Author
Hojjat, Seyed Ali

Publication Date
2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

New Models and Mechanisms for the Planning and Allocation of Online Advertising

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Management

by

Seyed Ali Hojjat

Dissertation Committee:
Professor John G. Turner, Chair
Professor Rick K.C. So
Professor Amelia C. Regan

2016

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ALGORITHMS

Page

ACKNOWLEDGMENTS

Words cannot begin to describe my gratitude and respect for my advisor, Dr. John Turner. He has been an unbelievably resourceful advisor and a remarkable mentor, a man of high standards, a true co-author, a best friend, the smartest and most diligent yet the kindest and most altruistic person I have met in my lifetime; I am forever honored to have been his student. This thesis would not have been possible without his guidance and encouragement. Our countless hours of meeting has not only benefited my knowledge and work, but has been the number-one reason for my skin to get some sunshine and for my face to get a shave (for which my wife is also thankful!).

Thanks to my other committee members, Dr. Rick So and Dr. Amelia Regan, for their evaluation of this manuscript and their valuable suggestions and advice; and to my extended advancement committee., Dr. Carlton Scott and Dr. Luyi Gui, for their encouragement in expanding my work in online advertising.

The idea and initial model for the first chapter of this thesis were formed during my internship at Yahoo Labs in Summer 2013. I am thankful to Jian Yang and Suleyman Cetintas for their valuable input and mentorship, granting me access to the wonderful dataset which I used for computational tests, and continued collaboration on the project.

I am thankful to all ODT faculty at the Merage School of Business (Professors Robin Keller, Carlton Scott, Rick So, Shuya Yin, Luyi Gui, and John Turner) for creating a remarkably peaceful and warm learning environment, and for their strong support and generous advice throughout my studies, particularly during my job hunting days. Also, thanks to Noel Negrete for her selfless care for Ph.D. students and service to the program.

I am eternally grateful to my parents, Mojgan and Kazem, for being the harshest and most brutally honest critics of my work since infancy, and for giving me attention in exchange for nothing but perfectionism and intellectual excellence! Big thanks to my grandparents, Zari and Fereydoon, for their everlasting love that always makes me feel like the star of *Everybody Loves Raymond*; to my brother, Arash, for being my partner-in-crime in impersonating members of the family; and to my Uncle Farid and Pargol who have cheered for each and every one of my scholarly achievements.

To my closest and dearest friends who have significantly impacted my character and philosophy of life: Saber Seyedali, Alireza Fatollahi, Amirhossein Kadivar, and Soroush Norouzi; many other friends with whom I have shared the best and sweetest memories in the past few years (Sepehr, Morteza and Mahboobeh, Mahdieh and Mahdi, Amir and Nasrin, Mojtaba, Anahita Kh., Shabnam H., Payam R., Soroush R.B., Navid and Ellie, Amir E.Z., and Abbas); and my fellow doctoral peers and classmates at the Merage School of Business for their friendship and support (Vahid, Ali E., Ali H.K., Yuhong, Candice, Jiaru, Yiwei, Yuhan,

Rico, Harsh, and Ran); I wish you all happiness and great success in your careers.

Finally, special thanks to my dear partner-in-life, Neda Masoud, for her love, attention, support, encouragement, believing in me, and pushing me to accomplish more and excel in my work. Being close to her was the reason I applied abroad and got my graduate degrees in the U.S.; I cannot imagine how my life would have been otherwise. I admire her diligence and devotion to academic research, and I wish her all the success at the University of Michigan.

S. Ali Hojjat, 2016

CURRICULUM VITAE

Seyed Ali Hojjat

EDUCATION:

Ph.D., Management (Operations and Decision Technologies) Sept. 2016
Paul Merage School of Business, University of California Irvine, Irvine, CA
Dissertation: "New models and mechanisms for the planning and allocation of online advertising"
Advisor: Dr. John Turner

M.B.A., Operations Management May 2011
Charlton College of Business, University of Massachusetts Dartmouth, North Dartmouth, MA
Thesis: "Pricing of private labels under different supply chain configurations"
Advisor: Dr. Soheil Sibdari

B.Sc., Industrial Engineering July 2009
Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran
Thesis: "A simulation approach to minimizing file fragmentation in disk allocation"
Advisor: Dr. Hashem Mahlooji

RESEARCH INTERESTS:

Applied Operations Research; Online Advertising; Revenue Management and Dynamic Pricing; Business Analytics; Machine learning; Data-driven Optimization; Social Networks; Game Theory; Supply Chain Management.

TEACHING INTERESTS:

Management Science; Revenue Management; Statistics; Operations Management; Supply Chain Management; Business Analytics; Linear /nonlinear programming; Large-scale optimization; Network models and algorithms.

RESEARCH:

- Hojjat, A., J. Turner, S. Cetintas, and J. Yang (2016). A unified framework for the scheduling of guaranteed targeted display advertising under reach and frequency requirements. Invited for 3rd round of review, *Operations Research*.
- Hojjat, A., S. Sibdari (2016). The Pricing of Private Labels in Different Supply Chain Configurations, Under review, *Naval Research Logistics*.
- 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.

CONFERENCE PRESENTATIONS:

- **Planning of Online Advertising:**
  - INFORMS 2015 Annual Meeting, Philadelphia, PA (Track TB45) Nov. 2015
  - INFORMS Revenue Management and Pricing Conference (Columbia University) June 2015
  - Southern California OR/OM Day (UCLA Anderson School of Business) May 2015
  - Paul Merage School of Business Annual Research Fest (poster) Apr. 2015
  - INFORMS 2014 Annual Meeting, San Francisco, CA (Track SC19) Nov. 2014
  - Paul Merage School of Business Annual Research Fest (poster) Apr. 2014
- **Pricing of Private Labels:**
  - INFORMS 2011 Annual Meeting, Charlotte, NC (Track WB67) Nov. 2011
  - INFORMS Northeast Regional Conference (UMass Amherst) May 2011

TEACHING EXPERIENCE:

- **Instructor:**
  - Business Statistics (Undergrad Core), UC Irvine Summer 2015
- **Recitation Leader and Head Teaching Assistant:**
  - Management Science (Undergrad & MBA core), UC Irvine F'12,'13, W'12,'14, S'13,'14,'15
  - Revenue Management (Undergrad & MBA elective), UC Irvine S'12, F'12,'13,'14,'15
  - Statistics for Management (MBA, EMBA, HCMBA core), UC Irvine W'13,'15
- **Teaching Assistant**
  - Predictive Analytics (Online, FEMBA elective), UC Irvine S'16
  - Supply Chain Management (Undergrad elective), UC Irvine S'12
  - Business Statistics (Undergrad core), UMass Dartmouth F'09, Sum'10, S'10,'11
  - Advanced Operations Management (MBA core), UMass Dartmouth F'09,'10, S'11
  - International Supply Chain Management (MBA elective), UMass Dartmouth S'11
  - Quantitative Business Analysis (Undergrad elective), UMass Dartmouth Sum'10

HONORS & AWARDS:

- Outstanding Teaching Assistant Award, elected by the Full Time MBA and Healthcare MBA classes of 2016, UC Irvine, June 2016.
- Outstanding Teaching Assistant Award, elected by the Full Time MBA Class of 2015, UC Irvine, June 2015.
- Ray Watson Doctoral Fellowship Award, Paul Merage School of Business, UC Irvine, May 2015
- Yahoo Invention Award, for an algorithm for delivering guaranteed display ads under reach and frequency requirements, IPCOM000237804D (with John Turner and Jian Yang), June 2014
- Joining Yahoo Labs (Sunnyvale, CA) as Intern Scientist, Summer 2013
- Four-year Doctoral Student Fellowship, Paul Merage School of Business, UC Irvine, Sept. 2011

ABSTRACT OF THE DISSERTATION

New Models and Mechanisms for the Planning and Allocation of Online Advertising

by

Seyed Ali Hojjat

Doctor of Philosophy in Management

University of California, Irvine, 2016

Professor John G. Turner, Chair

Motivated by recent trends in online advertising and advancements made by online publishers, most of this dissertation is devoted to the introduction, modeling, and the design of efficient allocation techniques for a new form of online advertising contract, which we refer to as *Reach and Frequency* (R&F) contract.

In the first chapter, we consider a type of R&F contract which allows online 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*) for him/her to be counted as reached. 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 are not present in existing models. 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, conducted on real industry data obtained from *Yahoo*, 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.

In the second chapter, we consider different variants of R&F contracts in which the advertiser specifies frequency using a probability distribution that details what fraction of users should see the ad how many times. This is a generalization of the R&F contract modeled in Chapter 1 in which frequency is a unique exposure count that every user should attain. Depending on whether the frequency of ad exposures to each user is measured over a fixed timespan (e.g., the number of times each user has seen the ad throughout the campaign's horizon) or on a rolling basis (e.g., over any randomly-selected 24-hour period), we propose two Markov chain models for serving ads and investigate how well they perform in maintaining a desired frequency distribution for an online ad campaign. We show that, when certain feasibility criteria are met, the publisher's impression assignment rule can be obtained very efficiently in linear time in the length of the frequency distribution specified by the advertiser.

The third and last chapter of this dissertation is concerned with the more basic problem of planning guaranteed targeted display advertising (GTDA), without the complication of R&F contracts. We examine three distinct lines of research: (1) Offline deterministic models that produce a plan based on mean supply forecasts, (2) An offline stochastic programming model which we develop as an intermediary benchmark, and (3) Real-time heuristics based on variants of online bipartite matching which require no supply forecast.We provide a brief review of the literature in each category and compare the performance of different approaches using simulation. We find that an online algorithm can outperform offline models (deterministic or stochastic) when the supply forecasts are even moderately noisy. In our simulations, we find that a specific bid-scaling function, not studied in the literature before, consistently outperforms other (well-studied) scaling functions. Using primal-dual analysis, we derive the competitive ratio of this scaling function and explain why and when it beats the best known bound of $1 - 1/e \simeq 0.63$.

Each of the three chapters in this dissertation is formatted as a separate research article, and therefore, bibliography and appendices are provided separately for each chapter.

CHAPTER 1:

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

1.1 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 $59.6 billion in 2015, exceeding all other forms of advertising such as broadcast and cable television, radio, newspaper, and consumer magazines (IAB 2016). 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

[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).

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 were 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 catchphrases 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 100% 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 on 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 their 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 as delivery, that pre-generates explicit sequence of ads for each user to plan his/her serving 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 novel and fresh application of column generation in in the form of an iterative algorithms 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 and 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 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 §1.2. In §1.3, 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 §1.4, we formally introduce how patterns can be used to serve advertising and describe our Pattern-G heuristic. In §1.5, 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 CG without hierarchical decomposition. Finally, we conduct a thorough set of numerical experiments in §1.6 to demonstrate the performance and robustness

of our methodology. Concluding remarks, insights and directions for future research appear in §1.7. Proofs of all theorems along with several extensions of the model and supplementary discussions are included in the appendices.

1.2 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 faster 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 basis 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.

1.3 The Ad Allocation Problem

The Reach & Frequency (R&F) ad planning problem we study is an allocation problem in which ads are assigned to individual users over a fixed time period (e.g., one week). Each advertiser 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). The publisher receives no payment for impressions shown to any user short of f_k. Additional exposures beyond f_k also result in no extra payment. Moreover, each advertiser wants only users from specific demographics; this is known as targeting. Over the planning period, many individuals of each demographic arrive, and each individual makes one or more visits and can be exposed to multiple impressions from one or more advertisers. It is a challenge to simultaneously satisfy all of the requirements from all advertisers; consequently, our goal is to maximize the quality of the ad allocation, which is

measured at both aggregate and disaggregate levels.

Maximizing aggregate quality is our primary goal, since it is usually tied to contractual obligations that have direct revenue consequences. For example, it is common for the publisher to consider revenue from guaranteed ads booked in advance, with any shortfalls in satisfying the reach targets r_k to be credited to the advertiser at the make-good cost rate c_k. Thus, if advertiser k receives u_k less than the r_k individuals she wishes to reach, the publisher pays an under-delivery penalty $c_k u_k$. Consequently, we can maximize retained revenue by maximizing the aggregate quality metric $-\sum_k c_k u_k$, or equivalently, minimizing the total cost of under-delivery $\sum_k c_k u_k$. This is perhaps the simplest aggregate quality metric possible. More complex aggregate quality metrics are often used in practice, e.g., to measure whether exposures are well-spread across different demographics; we will introduce and motivate such an aggregate quality metric shortly.

To provide users with a high-quality experience, as well as to provide advertisers with a high quality of ad delivery, the publisher also wishes to maximize the quality of the ad allocation at a disaggregate, or individual user, level. This can be achieved by, for example, making sure 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. Such user-level objectives constitute secondary goals, and the extent to which these secondary goals are met can be thought of as the ad plan's disaggregate quality. Although desired, disaggregate quality is typically not explicitly managed by existing ad serving systems. We will later formalize how we model disaggregate quality, and illustrate how specific user-level goals such as (i), (ii), and (iii) can be implemented.

The R&F ad planning problem thus defined has a primary objective that maximizes aggregate quality, and a secondary objective that maximizes disaggregate quality. 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.

This modeling choice captures the extent to which aggregate quality is more important than disaggregate quality, and is in line with the spirit of preemptive (lexicographic) goal programming (see Jones and Tamiz 2010).

The aggregate quality metric we use in our model incorporates not only under-delivery penalties as we have described above, but also measures how well-spread ads are across different demographics. As described more fully in Ghosh et al. (2009), advertisers prefer a *representative* allocation that shows ads to all demographics the advertiser chooses to target, yet spreads ads so that larger demographics receive a proportionally larger number of ads than smaller demographics. By requesting a representative allocation, an advertiser ensures the publisher does not fulfill the 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. We prefer our aggregate quality metric to include a quadratic non-representativeness penalty, because in conjunction with the specific constraints in our formulation, it allows us to write the primal solution as a closed-form expression of the dual solution; 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 if we have a way to recover the missing part of the primal solution that corresponds to this user's demographic. When the aggregate quality metric is quadratic, the dual solution we already have can be used to compute a near-optimal primal solution. On the other hand, when the aggregate quality metric is linear, this mapping between dual and primal solutions does not exist, and we say the allocation plan is not generalizable.

In what follows, we develop a model that solves the R&F ad planning problem by pre-generating explicit sequences of ads, which we call *patterns*, and then assigns these patterns to

specific users. In this model, aggregate quality is a summary statistic computed from the set of assigned patterns, and disaggregate quality is the sum total of the quality of ad sequences in the assigned patterns. Our proposed method, which we name Pattern-based Hierarchical Column Generation (Pattern-HCG), 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 our Pattern-HCG method 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, their objective function nicely combines a linear under-delivery penalty with a quadratic measure of non-representativeness, as discussed above, and is able to produce generalizable plans. As well, 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. Because of the structural similarity, we find it convenient and instructive to begin in §1.3.1 by describing their model while introducing our basic notation. 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 stage-one math program would be small enough to solve using a commercial solver on a single machine without SHALE. Next, in §1.3.2 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. A list of mathematical notation is provided in Appendix 1.A for quick reference.

1.3.1 Allocation of Impression-based Ad Campaigns

A typical method used to plan and serve impression-based ads is to solve an optimization problem that matches forecasted supply with campaigns' demand and produces a short-term allocation plan (*offline* phase), and then use the resulting policy for assigning user impressions to different ad campaigns at serving time (*online* phase). The offline optimization problem is re-solved periodically to update the policy with adjusted 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) 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 1.1 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

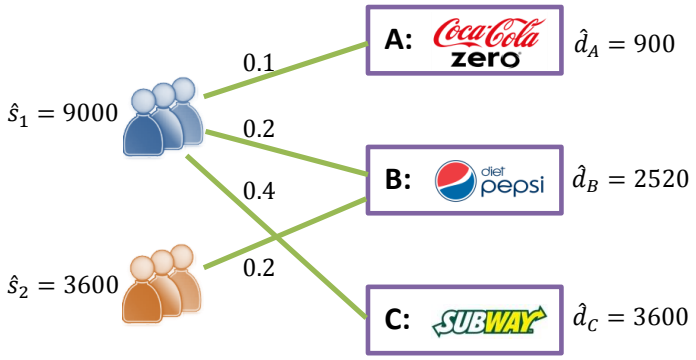


Figure 1.1: Example Bipartite Graph with Impression-based Ad Campaigns

the caret (ˆ) to differentiate between quantities that we measure as a number of 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:

$$\text{(IA):} \quad \text{Minimize:} \quad \sum_{k, i \in \hat{\Gamma}(k)} \frac{\hat{s}_i}{2\hat{\theta}_k} \hat{w}_k \left(\hat{x}_{ik} - \hat{\theta}_k \right)^2 + \sum_k \hat{c}_k \hat{u}_k \tag{1.1a}$$

$$\text{s.t.} \quad \sum_{i \in \hat{\Gamma}(k)} \hat{s}_i \hat{x}_{ik} + \hat{u}_k \geq \hat{d}_k \qquad \forall k \tag{1.1b}$$

$$\sum_{k \in \hat{\Gamma}(i)} \hat{x}_{ik} \leq 1 \qquad \forall i \tag{1.1c}$$

$$\hat{x}_{ik},\ \hat{u}_k \geq 0 \qquad \forall i, k \tag{1.1d}$$

Demand constraint (1.1b) 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 (1.1c) states we cannot allocate more than 100% of supply from each node i. The objective function (1.1a) 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. A perfectly-representative allocation is defined as one that distributes the demanded impressions of every campaign uniformly across its total eligible supply, i.e., each campaign k grabs a $\hat{\theta}_k = \hat{d}_k/\hat{S}_k$ proportion of each eligible supply pool $i \in \hat{\Gamma}(k)$. This has the interpretation that every serving opportunity eligible for campaign k is treated the same way, with campaign k winning the impression with constant probability $\hat{\theta}_k$. Since the perfectly-representative allocation is often infeasible to achieve for all campaigns, deviations from this ideal are quadratically penalized in the objective. Weighting the terms by $\hat{s}_i/2\hat{\theta}_k$ is for mathematical convenience and balancing the relative magnitude of the terms in the objective. 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 1.1 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 1.1 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.

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 §1.5.1, we extend SHALE to solve the aggregate reach planning component of our R&F allocation problem.

1.3.2 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 1.1.

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

Algorithm 1.1 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': $q_{jk'} \leftarrow q_{jk'} + 1$.

in current practice, defines the *maximum* number of exposures beyond which the publisher will no longer receive a payment. Indeed, our numerical experiments in §1.6 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 reasonable 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.

1.4 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. Such a setup is becoming quite common, e.g., in video advertising as well as ads which are filled as the user scrolls down the webpage. In Appendix 1.C we discuss many practical use cases for one-dimensional patterns even when the publisher's page has multiple advertising positions, and we extend our model to handle the case of two-dimensional patterns which explicitly plan multiple ads for each user visit.

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 1.2 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 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

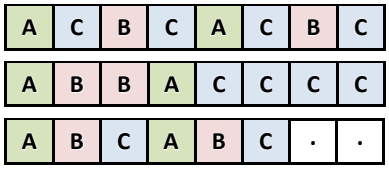


Figure 1.2: Examples of patterns with three campaigns {A,B,C}

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 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 §1.6 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 1.E 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 §1.5 we will show how we pre-generate and then serve optimal patterns using our Pattern-HCG method.

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

Recall from §1.3.1 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:

$$\text{(RA):} \quad \text{Minimize:} \quad \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}}{2\theta_k} w_k \left(x_{vik} - \theta_k\right)^2 + \sum_k c_k u_k \tag{1.2a}$$

$$\text{s.t.} \quad \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} + u_k \geq r_k \qquad \forall k \tag{1.2b}$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq 1 \qquad \forall v,i \tag{1.2c}$$

$$0 \leq x_{vik} \leq 1 \qquad \forall v,i,k \in \Gamma(v,i) \tag{1.2d}$$

$$u_k \geq 0 \qquad \forall k \tag{1.2e}$$

Demand constraint (1.2b) 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.

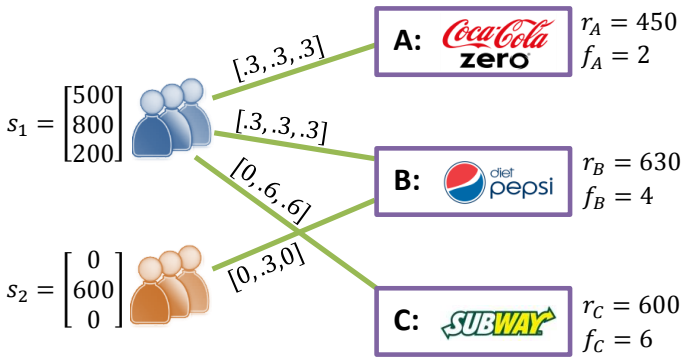
Supply constraint (1.2c) is structurally different from its counterpart (1.1c) in (IA). A naïve translation of (1.1c) yields $\sum_{k\in(v,i)} x_{vik} \leq 1$. However, we can immediately see that such a constraint would be too strict. Indeed, if campaigns A and B each require only one impression (i.e., $f_A = f_B = 1$), 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 (1.2c) 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 (1.2c) does not imply $x_{vik} \leq 1$ as its counterpart (1.1c) in (IA) did, we now explicitly

enforce the upper-bounds $x_{vik} \leq 1$ using constraint (1.2d) to ensure x_{vik} can be interpreted as a proportion.

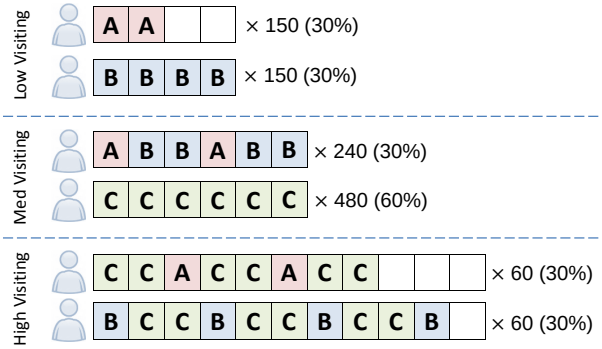
Figure 1.3 provides a solution to an instance of (RA), as well as one possible extension of this solution to specific patterns. In this example, the publisher receives visits from $s_1 = 1500$ unique individuals of demographic 1, of which $\{500, 800, 200\}$ users are classified as {low,med,high}-visiting, 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 1.1.

In Figure 1.3(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 1.3(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



(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 1.3: Example Bipartite Graph and Pattern-Based Solution of R&F Campaigns

Algorithm 1.2 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.

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, $\{200, 80, 80\}$ 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 1.2, 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 §1.5.

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 Appendix 1.D we present a more advanced greedy algorithm which simultaneously does a better job of packing R&F ad impressions into patterns and maintaining the reach fractions x^*_{vik}. In the following section, we introduce a method which explicitly considers the packing problem of pattern generation, and pre-generates optimal patterns.

1.5 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 $k \in K$ 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 1.K). 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 is the standard goal 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. In our case, this corresponds to the number of campaigns, thus the diversity of ads served to a user which 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 1.F). 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 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.

1.5.1 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:

$$\text{(RA-}\delta\text{):} \quad \text{Minimize} \quad \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}}{2\theta_k} w_k \left(x_{vik} - \theta_k\right)^2 + \sum_k c_k u_k \qquad \underline{\text{Duals}} \text{ (All} \geq 0) \tag{1.3a}$$

$$\text{s.t.} \quad \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} + u_k \geq r_k \qquad \forall k \qquad \alpha_k \tag{1.3b}$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq \delta_{vi} \qquad \forall v,i \qquad \beta_{vi} \tag{1.3c}$$

$$0 \leq x_{vik} \leq 1 \qquad \forall v,i,k \in \Gamma(v,i) \qquad \gamma^L_{vik}, \gamma^U_{vik} \tag{1.3d}$$

$$u_k \geq 0 \qquad \forall k \qquad \varphi_k \tag{1.3e}$$

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 (1.3c) is a generalization of our earlier supply constraint (1.2c) 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 §1.5.4, 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 1.G.

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

ships:

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.*

In Algorithm 1.3, 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.,

Algorithm 1.3 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 $\alpha_k = c_k$.

reach) r_k is satisfied. We provide a generalized proof of convergence in Appendix 1.I which does not rely on all α_k values being initialized at 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[2]. 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.*

[2]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.

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.

1.5.2 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):

$$\text{(PA):} \quad \Psi_{vi} := \text{Minimize} \quad \sum_{p \in \mathcal{P}_{vi}} \pi_{vip} y_{vip} \qquad \underline{\text{Duals:}} \tag{1.4a}$$

$$\sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip} = s_{vi} x^*_{vik} \quad \forall k \in \Gamma(v, i) \qquad \bar{\alpha}_{vik} \text{ (free)} \tag{1.4b}$$

$$\sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq s_{vi} \qquad \bar{\beta}_{vi} \geq 0 \tag{1.4c}$$

$$y_{vip} \geq 0 \quad \forall p \in \mathcal{P}_{vi} \qquad - \tag{1.4d}$$

Constraint (1.4b) 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 (1.4c) 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) which can be done in parallel.

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.

1.5.3 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 would be most beneficial to add to each pattern pool $\mathcal{P}_{vi}$. The reduced cost of the y_{vip} variable in (PA) is given by $\pi_{vip} - \sum_{k \in \Gamma(v,i)} \bar{\alpha}^*_{vik} b_{kp} + \bar{\beta}^*_{vi}$. Therefore, the pattern generation problem,

which constructs a new pattern for user type (v,i), has the following form:

$$\text{(PG):} \qquad \psi_{vi} := \text{Minimize} \quad \pi(\mathbf{b}) \; - \sum_{k \in \Gamma(v,i)} \bar{\alpha}^*_{vik} b_k \tag{1.5a}$$

$$\text{s.t.} \quad \sum_{k \in \Gamma(v,i)} f_k b_k \leq L_v \tag{1.5b}$$

$$b_k \in \{0,1\}, \quad \forall k \in \Gamma(v,i) \tag{1.5c}$$

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 (1.5b) 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 are able to solve more than 1000 such knapsack problems[3] 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

[3]This runtime corresponds to the problem instances we study in §1.6, which have $|\Gamma(v,i)|$ between 1 and 442 with an average connectivity of 36 campaigns per viewer type, and three pattern lengths $L_v \in \{10, 19, 56\}$.

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 could 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 1.B.

1.5.4 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):

$$\text{(PA-F):} \qquad \Psi_{vi}^{(F)} := \text{Minimize} \quad \sum_{p \in \mathcal{P}_{vi}} y_{vip} \qquad \text{Duals:} \tag{1.6a}$$

$$\sum_{p \in \mathcal{P}_{vi}} b_{kp} y_{vip} = s_{vi} x_{vik}^* \quad \forall k \in \Gamma(v,i) \qquad \bar{\alpha}_{vik}^{(F)} \text{ (free)} \tag{1.6b}$$

$$y_{vip} \geq 0 \quad \forall p \in \mathcal{P}_{vi} \qquad - \tag{1.6c}$$

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 (1.4c) 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 $\bar{\alpha}_{vik}^{*(F)} = 1$, $\forall k \in \Gamma(v, i)$. 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:

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

If (PG-F) concludes with $\psi_{vi}^{*(F)} < 0$, 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 $\Psi_{vi}^{*(F)} > s_{vi}$, 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. In this case, 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:

$$\sum_{p\in\mathcal{P}_{vi}} \left(\sum_{k\in\Gamma(v,i)} f_k b_{kp} \right) y_{vip}^* = \sum_{k\in\Gamma(v,i)} f_k \left(\sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip}^* \right) =^{(1.6b)}= \sum_{k\in\Gamma(v,i)} f_k s_{vi} x_{vik}^*.$$

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 $\Psi_{vi}^{*(F)}$ users provides L_v impressions, the effective impression utilization rate at the optimal solution to (PA-F) is given by $\sum_{k\in\Gamma(v,i)} \frac{f_k s_{vi} x_{vik}^*}{L_v \Psi_{vi}^{*(F)}}$. Based on this analysis, we suggest the following update rule:

$$\delta_{vi} \;\leftarrow\; s_{vi} X_{vi}^* / \Psi_{vi}^{*(F)} - \epsilon, \tag{1.7}$$

where $X_{vi}^* = \sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik}^*$ is the left-hand side of constraint (1.3c) 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 $\Psi_{vi}^{*(F)} \leq s_{vi}$, 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 $\pi(\cdot)$ is not linear, however, in the pattern improvement phase we no longer solve the large-scale math program (RA-δ). Again, we remind the reader that iterations between (PA-F) and (PG-F), or (PA) and (PG), can be conducted in parallel 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 1.4.

Remark 1: The value of δ_{vi} always decreases following update rule (1.7). This follows since $X^*_{vi} \leq \delta_{vi}$ due to constraint (1.3c), 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 1.K for a proof of this behavior in the more general case of the cutting stock problem.

Algorithm 1.4 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 1.3)
  - 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 (1.7).
  - 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 1.2) 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 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 1.J.

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.

1.6 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 §1.3.2, 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 §1.4.1, 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%.

1.6.1 Data

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 $\mathcal{V} = \{\text{low,med,high}\}$-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[4], 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, with larger numbers given lower weights. 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.

1.6.2 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

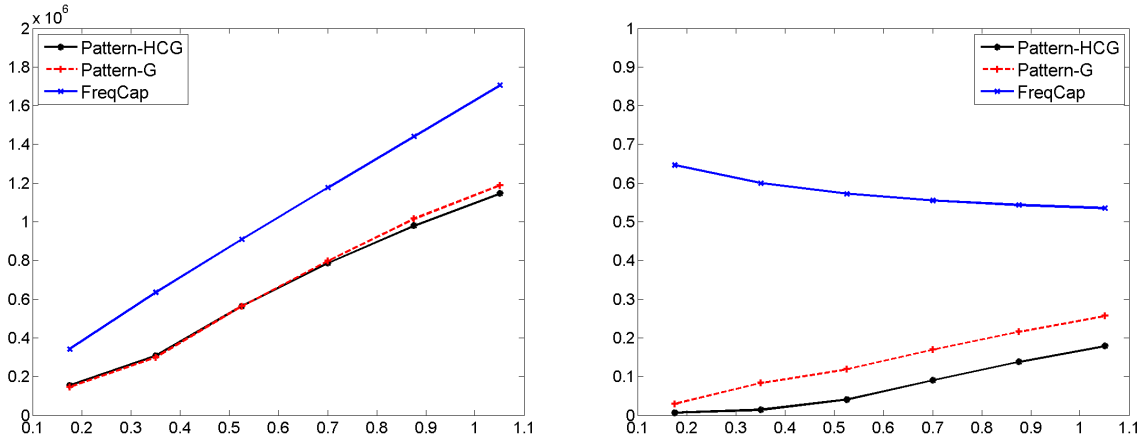
[4]We use week 4 as it gave us a slightly higher number of supply nodes with $s_{vi} > 0$, i.e., a more complete graph, compared to other weeks.

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). More details about each test and the results appears below.

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 $\mathcal{S}_{R\&F} = 0.43\mathcal{S}_{Tot}$. 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 1.4 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



(a) Non-representativeness Penalty

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

Figure 1.4: Performance of our three methods at different levels of sellthrough $\mathcal{S}_{R\&F}$.

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

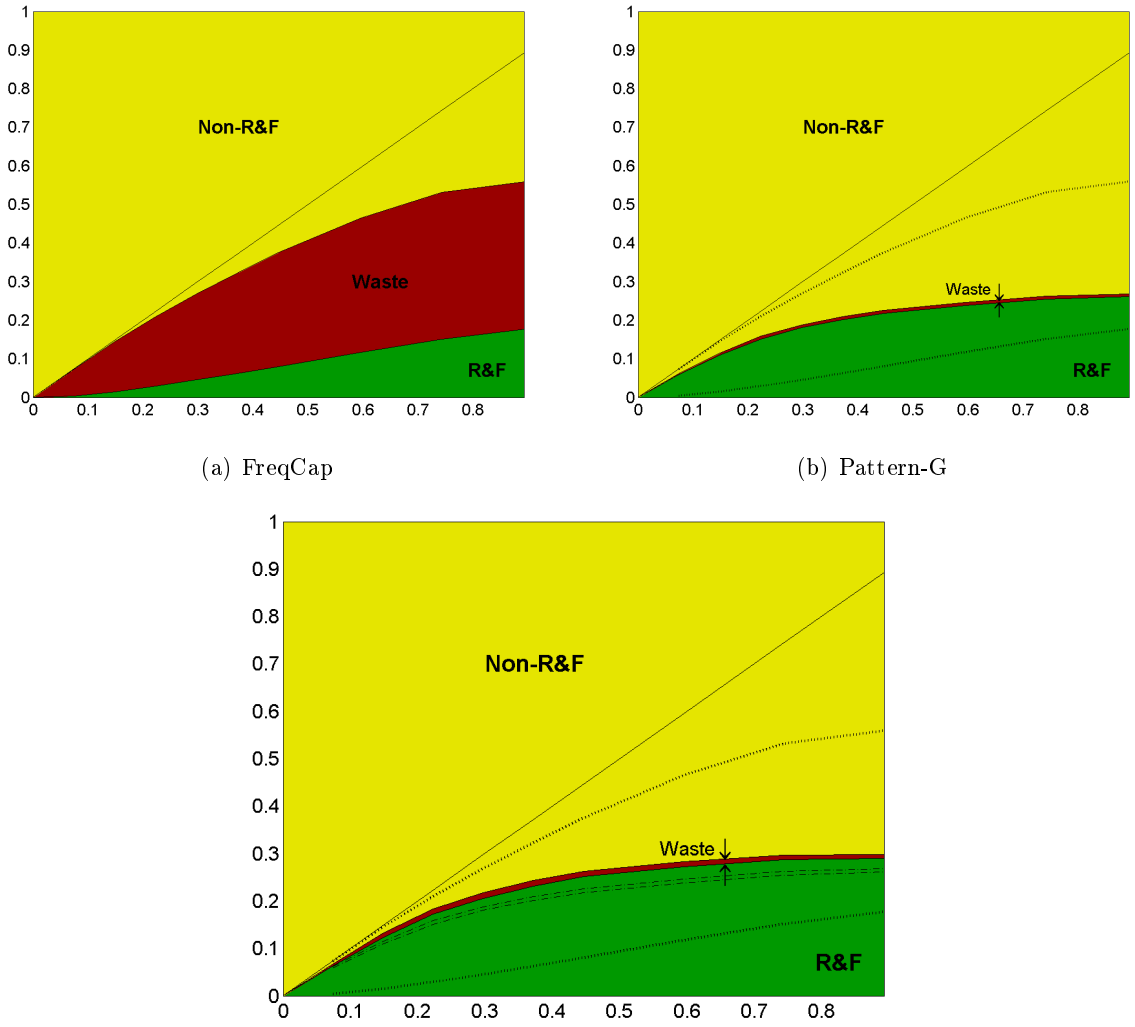
Figure 1.5(a) shows that FreqCap allocates the most number of impressions to R&F campaigns, but nearly 2/3 of these impressions fall short of the target frequency at the

user level, and therefore end up as waste. Figure 1.5(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 1.5(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 1.4).

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[5] 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 $\phi_v(\cdot)$

[5]That is, we set $s_{vi}^{(a)} \leftarrow s_{vi}^{(a)} \big(\sum_{v',i'} s_{v'i'} \big) / \big(\sum_{v',i'} s_{v'i'}^{(a)} \big)$.

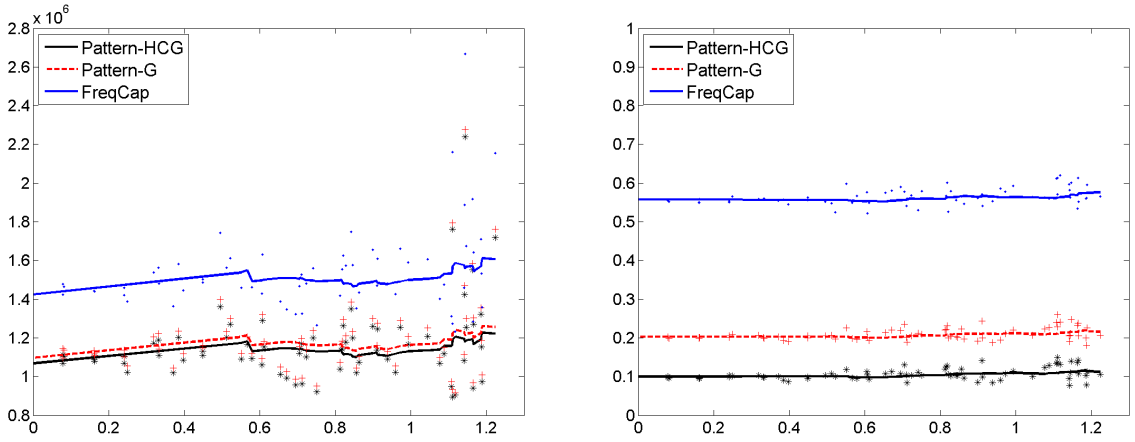


(a) FreqCap

(b) Pattern-G

(c) Pattern-HCG

Figure 1.5: Comparing the ratio of wasted traffic across different solution algorithms 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.



(a) Non-representativeness Penalty

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

Figure 1.6: Performance under noisy forecasts, as a function of mean absolute percentage error (MAPE).
Each dot corresponds to a different random arrival stream.

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 1.6 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[6]. The curves are basic moving averages which illustrate the overall trend.

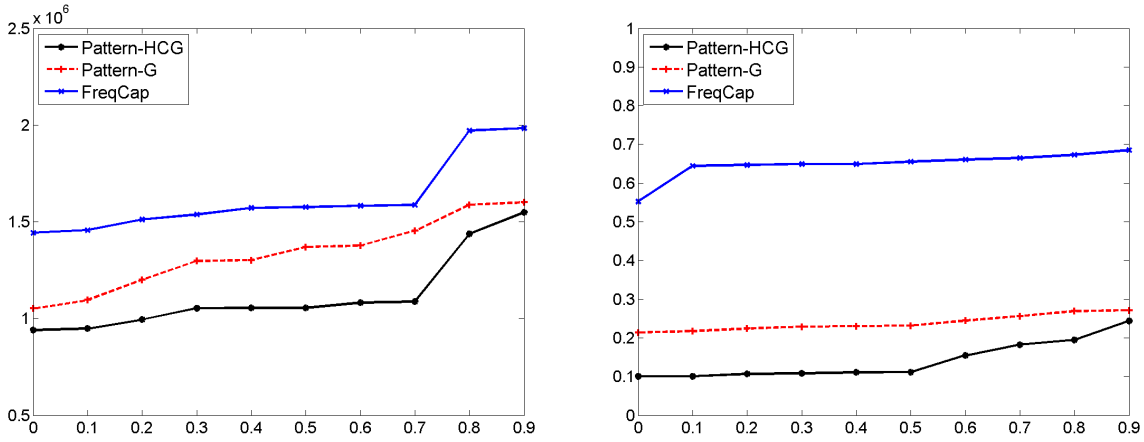
[6]The assignment of patterns to users is a random process (pattern p is chosen for a user of type (v, i) with probability y^*_{vip}/s_{vi}) and differs in each run of the simulation, which has a slight impact on the performance. For each arrival stream, the solution was simulated multiple times to accurately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

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.

Test 3: Robustness to Graph Sampling & Generalizability

As described in §1.3, 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 1.7 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



(a) Non-representativeness Penalty

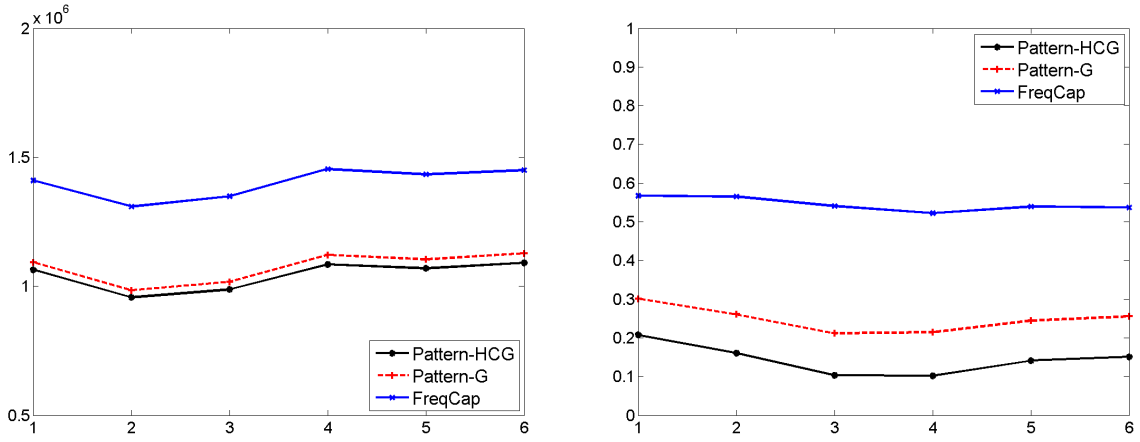
(b) Under-delivery Fraction $\sum_k u_k / \sum_k r_k$

Figure 1.7: Performance in the presence of generalized arrivals.
The horizontal axis shows the fraction of supply nodes omitted from the graph at optimization/planning time.

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 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 1.8. 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.



(a) Non-representativeness Penalty

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

Figure 1.8: Out of sample testing
Performance measured in weeks 1-6, using data from week 4 for parameter estimation and optimization.

We observe that the relative performance gaps are consistent among the three methods across all 6 weeks, with Pattern-HCG consistently performing best.

In fact, the results in Figure 1.8 match our findings in Tests 1 and 2. We found that the MAPE measure between the s_{vi} of week 4 and any of the other 5 weeks is consistently close to 37%. From Figure 1.6(b), we expect the effect of forecast noise to be negligible at a MAPE of 37%. However, we find the aggregate supply of users to fluctuate across the 6 weeks, with $\mathcal{S}_{R\&F} = \{1.01, 0.94, 0.87, 0.87, 0.91, 0.94\}$, respectively. It is easy to see that the direction and magnitude of changes in under-delivery levels in Figure 1.8(b) is closely related to the change in sellthrough level $\mathcal{S}_{R\&F}$. We should expect this behavior due to the linear relationship observed in Figure 1.4(b) in the high-sellthrough domain. It is harder to comment on changes in non-representativeness penalty. At a MAPE of 37%, we know from Figure 1.8(a) that non-representativeness can show moderate variation depending on the random instance of $s_{vi}^{(a)}$. The most assuring observation in Figure 1.8 should be the consistency among relative performance of the three methods across isolated real-life instances of the user arrival stream.

1.7 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 1.E, 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.

Finally, we note that our pattern-based approach for serving web advertisements can also be applied to other forms of technology-enabled advertising, including digital TV, online videos, and in-game advertising.

Bibliography

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`.

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.

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.

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.

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 (2016, April). IAB 2015 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.

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

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.

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* .

Appendices

1.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. Users of visit-type v will visit at least L_v times and see 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 Appendix B.
δ_{vi}	Proportion of type-(v,i) impressions usable when serving with patterns (considering 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).

1.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 §1.5.3, (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 (1.5b), 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 $L_v s_{vi} - \sum_{k\in\Gamma(v,i)} f_k s_{vi} x^*_{vik}$. 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 1.E 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:

$$\pi_{pacing}(\mathbf{b}) \;=\; \text{Minimize} \quad \sum_{k\in\Gamma(v,i)} \sum_{\ell=1}^{L_v} \left(\sum_{\ell'=1}^{\ell} z_{k\ell'} - b_k T_\ell \right)^2 \tag{1.8a}$$

$$\sum_{\ell=1}^{L_v} z_{k\ell} \;=\; b_k f_k \qquad \forall k \in \Gamma(v,i) \tag{1.8b}$$

$$\sum_{k\in\Gamma(v,i)} z_{k\ell} \;\leq\; 1 \qquad \forall \ell = 1, ..., L_v \tag{1.8c}$$

$$z_{k\ell} \in \{0,1\} \tag{1.8d}$$

Constraint (1.8b) 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 (1.8c) 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 (1.8b)).

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 (1.8) in isolation but rather we wish to solve (1.8) as an inner-optimization within (PG). Unfortunately, we cannot transform (1.8) into an assignment problem when the $\mathbf{b}$ vector is also a decision variable. Instead, to integrate (1.8) into (PG), we use (1.8a) as the objective and include the constraints (1.8b,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 (1.8) 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 $T_\ell = \left(\frac{f_k}{L_v}\ell - I_k\right)^+$. 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:

$$\pi_{pacing}(\mathbf{b}) \;=\; \text{Minimize} \quad \sum_{k} \sum_{j_k=2}^{f_k} \left| Z_{j_k} - Z_{(j-1)_k} - \frac{L_v}{f_k} b_k \right| \tag{1.9a}$$

$$\sum_{j_k=1}^{f_k} \sum_{\ell=1}^{L_v} z_{j_k \ell} \;=\; f_k b_k \qquad \forall k \tag{1.9b}$$

$$\sum_{k} \sum_{j_k=1}^{f_k} z_{j_k \ell} \;\leq\; 1 \qquad \forall \ell = 1, ..., L_v \tag{1.9c}$$

$$Z_{j_k} \;=\; \sum_{\ell=1}^{L_v} \ell z_{j_k \ell} \qquad \forall k,\; j_k = 1, ..., f_k \tag{1.9d}$$

$$Z_{j_k} \;\geq\; Z_{(j-1)_k} + 1 \qquad \forall k,\; j_k = 2, ..., f_k \tag{1.9e}$$

$$z_{j_k \ell} \in \{0, 1\}, \quad Z_{j_k} : \text{Integers} \tag{1.9f}$$

Constraints (1.9b) and (1.9c) perform the same function as (1.8b) and (1.8c). Constraint (1.9d) establishes the relationship between variables $z_{j_k \ell}$ and Z_{j_k}, and constraint (1.9e) 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 (1.9), 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 (1.9) as well 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 (1.9a) as the objective and (1.9b-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 (1.8) rather than (1.9).

4. Competing campaigns

Campaigns of competing brands may target similar user demographic,s, 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.

Alternatively, let $C_k \subseteq \mathcal{K} \backslash \{k\}$ denote the set of competing campaigns specified by advertiser k. Including a constraint for the form $\sum_{k' \in C_k} b_{k'} \leq |C_k|(1 - b_k)$ in (PG) ensures that once campaign k is included in the pattern ($b_k = 1$), then no competing brand is included ($\sum_{k' \in C_k} b_{k'} = 0$).

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.

1.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 1.2 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.[7]

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

[7] Decision variables in (PA) need to be updated accordingly to y_{vihp} to denote the number of times a pattern $p \in \mathcal{P}_{vih}$ should be assigned to users of type (v, i) in position h. The left-hand side of constraint (??c) should be further divided by H, due to the fact that the impression supply of user type (v, i) is now HL_vsvi rather than L_vsvi. The rest of the model remains unchanged.

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 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 1.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 1.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.

We should also point out that if re-exposing the user to the same ad is meant to increase brand awareness or entice a click, then it is not clear whether delivering impressions upon separate visits will be any more effective than showing multiple instances upon a single visit. For example, multiple instances of the same ad that turn around the regular coloring of a familiar page could draw a user's attention to a higher degree as opposed to showing one instance of the ad in the same position upon separate visits. Further empirical studies in this regard can provide useful insight as to whether it is justified for advertisers to explicitly require the frequency to be delivered across separate visits.

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 (1.3c), 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 1.B. Sub-problem (PG) can be cast as:

$$\text{Minimize} \quad \pi(\mathbf{b}) - \sum_{k \in \Gamma(v,i)} \bar{\alpha}^*_{vik} b_k \tag{1.10a}$$

$$\text{s.t.} \quad \sum_{k \in \Gamma(v,i)} f_k b_{kh} \leq L_v \qquad \forall h = 1, ..., H \tag{1.10b}$$

$$b_k \equiv \sum_{h=1}^{H} b_{kh} \leq 1 \qquad \forall k \in \Gamma(v,i) \tag{1.10c}$$

$$b_{kh} \in \{0,1\}, \qquad \forall k \in \Gamma(v,i),\ \forall h = 1, ..., H \tag{1.10d}$$

Constraint (1.10b) is analogous to (1.5b) 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 (1.10c) 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 1.B can be adapted to two-dimensional patterns. We would use $\pi_{excess}(\mathbf{b}) = (HL_v - \sum_k f_k b_k)\bar{p}_{vi}$. 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 (1.10) 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).

1.D An Improved Greedy Algorithm for Pattern Generation

Here we provide a more advanced greedy algorithm for pattern generating. The goal is to improve upon the packing of R&F ad impressions into patterns, while maintaining the optimal reach fractions x^*_{vik}. Similar to the Pattern-G algorithm introduce in §1.4.1, we start by solving the Reach Allocation (RA) problem. The difference lies in how we sample impressions. In the improved greedy heuristic, henceforth Pattern-G+, we employ a *fluid* sampling of one impression at a time, together with a queue (or stack) that holds all unused impressions sampled for a supply node.

Note that when a x_{vik} fraction of *user*-supply should be reached by campaign k, then a $\frac{f_k}{L_v} x_{vik}$ fraction of *impression*-supply should be allocated to each campaign k (see constraint (1.2c)), and a $1 - \sum_{k \in \Gamma(v,i)} \frac{f_k}{L_v} x_{vik}$ proportion of *impression*-supply should be allocated to *excess*. Therefore, to construct patterns for users of class (v, i), we randomly draw impressions of R&F contracts $k \in \Gamma(v, i)$ according to weights $\frac{f_k}{L_v} x_{vik}$ and we include *excess* as an additional campaign with sampling weight $1 - \sum_{k \in \Gamma(v,i)} \frac{f_k}{L_v} x_{vik}$. We store the sampled impressions in a queue list $\mathcal{Q}_{vi}$ in preserved order. Then we construct a pattern using R&F campaigns which have attained (at least) f_k impressions in the queue. Excess impressions in the queue are used in two ways: 1) to fill-in the slack when a pattern is closed, and 2) to construct a fully-blank pattern (when we can find L_v excess impressions in the queue). The former is important to maintain the allocation of impression supply determined by (1.2c), and the latter mimics $\sum_p y_{vip} \leq s_{vi}$ in the corresponding CG solution (i.e., allows some users to not be reached by any campaign at all).

A crucial part of this heuristic which determines its runtime and stability and is to employ the least amount of sampling so the impression queue $\mathcal{Q}_{wi}$ does not grow indefinitely and is used at about the same rate as new impressions are sampled. In our numerical experiments, we did not observe a case where the $\mathcal{Q}_{wi}$ would grow indefinitely. Figure 1.9 illustrates the

Algorithm 1.5 Improved Pattern-based Greedy Heuristic (***Pattern-G+***)

- **OFFLINE:**
  - Solve the reach allocation problem (RA).
  - For each supply node (v, i): Set $\mathcal{Q}_{vi} = \{\}$ (impression queue), $N_{vi} = 0$ (number of patterns), $\hat{L}_{vi} = L_v$ (remaining slots from the open pattern being constructed). $b_k = 0$, $\forall k \in \Gamma(v, i)$ (no campaign included in the open pattern yet).
- **ONLINE:** Upon the first visit from user j from of type (v, i):
  - If $\mathcal{Q}_{vi}$ is empty: Sample one random impression among contracts $k \in \Gamma(v, i)$ according to weights $f_k x^*_{vik}$ or an *excess* impression according to weight $L_v - \sum_{k \in \Gamma(v,i)} f_k x^*_{vik}$, and append to the queue $\mathcal{Q}_{vi}$.
  - PICK: Set k^*= index of the contract whose impression is at the head of the queue. (This contract should now be packed in the pattern). If k^* is an *excess* impression, set $q_{k*} = L_v$.
  - CHECK FIT:

    IF $f_{k^*} \leq \hat{L}_{vi}$ (k^* fits in the remaining slots of the pattern) and $b_{k^*} = 0$ (k^* is not already added to the pattern):
    - SAMPLE: If there are less than f_{k^*} impressions of k^* in $\mathcal{Q}_{vi}$:, sample random impressions (of contracts $k \in \Gamma(v, i)$ or excess, as described before) and append to $\mathcal{Q}_{vi}$ until exactly f_{k*} impressions of k^* are in the queue.
    - ADD TO PATTERN:
      Move the first f_{k^*} impressions of k^* from $\mathcal{Q}_{vi}$ into the pattern.
      Set $b_{k*} = 1$. Update $\hat{L}_{vi} \leftarrow \hat{L}_{vi} - f_{k^*}$.

    ELSE:
    - CLOSE PATTERN:
      Move $\hat{L}_{vi}$ *excess* impressions from $\mathcal{Q}_{vi}$ to complete the pattern
      (if not available, SAMPLE more impressions at random until $\hat{L}_{vi}$ excess impressions are present).
      Store pattern info in P_j and assign to user. Upon subsequent visits from a user: Randomly draw one impression from P_j to show to the user. Remove that impression from P_j.
    - START A FRESH PATTERN: Set $\hat{L}_{vi} = L_v$ and all $b_k = 0$, $\forall k \in \Gamma(v, i)$.

```
Q = {}, Open Pattern: [], Slack = 10
Q is empty, sampling 1 impression ...
Q = {C}, Open Pattern: [], Slack = 10
Picked contract "C" from Q. Needs 5 impressions to be reached.
Not enough impressions in Q. More sampling until achieved ...
Q = {CBA-B---CCAAACBC}, Open Pattern: [], Slack = 10
Moving 5 impressions of "C" into the current pattern.
Q = {BA-B---AAAB}, Open Pattern: [C], Slack = 5
Picked contract "B" from Q. Needs 2 impressions to be reached.
Moving 2 impressions of "B" into the current pattern.
Q = {A----AAAB}, Open Pattern: [BC], Slack = 3
Picked contract "A" from Q. Needs 7 impressions to be reached.
Not enough impressions in Q. More sampling until achieved ...
Q = {A----AAABCB---AACC-A}, Open Pattern: [BC], Slack = 3
Already exists in the open pattern or doesn't fit. Close the pattern: [BC]
Disposing 3 Blank impressions into the closed pattern to fill up slack.
Q = {A-AAABCB---AACC-A}, Open Pattern: [], Slack = 10
Picked contract "A" from Q. Needs 7 impressions to be reached.
Moving 7 impressions of "A" into the current pattern.
Q = {-BCB---CC-}, Open Pattern: [A], Slack = 3
...
After 99 Patterns are Generated: (Note that queue length remains stable)
...
Q = {CCCAAACA-CAAA--AAA--C-CC---}, Open Pattern: [], Slack = 10
Picked contract "C" from Q. Needs 5 impressions to be reached.
Moving 5 impressions of "C" into the current pattern.
Q = {AAAA-AAA--AAA--C-CC---}, Open Pattern: [C], Slack = 5
Picked contract "A" from Q. Needs 7 impressions to be reached.
Already exists in the open pattern or doesn't fit. Close the pattern: [C]
Disposing 5 Blank impressions into the closed pattern to fill up slack.
Q = {AAAAAAAAAAC-CC---}, Open Pattern: [], Slack = 10
```

Figure 1.9: Step-by-step demonstration of the Pattern-G+ heuristic.

step-by-step progress of our heuristic for a particular supply node with $s_{vi} = 100$, $L_v = 10$, $k \in \{A, B, C\}$, $q_k \in \{7, 2, 5\}$, $x^*_{viA} = 0.4$, $x^*_{viB} = 0.7$, $x^*_{viC} = 0.5$. The optimal solution obtained by column generation is: $40[AB] + 30[BC] + 20[C] + 10[\ ]$. The improved greedy heuristic produces the solution: $32[AB] + 32[BC] + 17[C] + 8[A] + 8[B] + 3[\ ]$ which gives the reach fractions $\{.40, .72, .49\}$ which are quite close to x^*_{vik}.

Note that our heuristic can also be used as an offline method for solving the general cutting stock problem. To this end, we would ask the algorithm to produce a large number of patterns; then we group identical patterns and produce a {pattern, usage frequency} solution, similar to the output obtained from CG. For *generalized* user arrivals in our online advertising

application, one may only ask for a single pattern from the heuristic and discard the $\mathcal{Q}_{wi}$, or construct and store a full (offline) solution when a new user type is observed for the first time.

1.E 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 §1.4 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 (1/3 of the impression traffic from this user type) is not considered for R&F planning (when $X \sim Poiss(30)$, $E[\max(0, X-20)] = 10.049$). 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 in browsing-behavioral class w 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 user of type v, 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:

$$\text{(PA-R):} \quad \Psi_{vi}^{(R)} := \text{Minimize} \quad \sum_{p\in\mathcal{P}_{vi}} \pi_{vip} y_{vip} \qquad \underline{\text{Duals:}} \qquad (1.11a)$$

$$\sum_{p\in\mathcal{P}_{vi}} \bar{b}_{kp} y_{vip} = s_{vi} x^*_{vik} \quad \forall k \in \Gamma(v,i) \qquad \bar{\alpha}_{vik}^{(R)} \text{ (free)} \qquad (1.11b)$$

$$\sum_{p\in\mathcal{P}_{vi}} y_{vip} \le s_{vi} \qquad \bar{\beta}_{vi}^{(R)} \ge 0 \qquad (1.11c)$$

$$y_{vip} \ge 0 \quad \forall p \in \mathcal{P}_{vi} \qquad - \qquad (1.11d)$$

where the optimal reach proportions x^*_{vik} from (RA-δ) are sought in expectation. The only change from (PA) in (PA) is the substitution of b_{kp} with $\bar{b}_{kp}$ in (1.4b). The pattern generating subproblem takes the following form:

$$\text{(PG-R):} \qquad \psi_{vi}^{(R)} := \text{Maximize} \quad \sum_{k\in\Gamma(v,i)} \bar{\alpha}_{vik}^{*(R)} \underbrace{\Big(\sum_{k=1}^{\bar{L}_v} \phi_v^{(\ell)} I_{k\ell}\Big)}_{\bar{b}_k} - \pi(\mathbf{b}) \tag{1.12a}$$

$$\sum_{k\in\Gamma(v,i)} z_{k\ell} \leq 1 \qquad \ell = 1, ..., \bar{L}_v \tag{1.12b}$$

$$\sum_{\ell=1}^{\bar{L}_v} z_{k\ell} = f_k b_k \qquad \forall k \in \Gamma(v,i) \tag{1.12c}$$

$$\sum_{\ell'=1}^{k} z_{k\ell'} \leq f_k - 1 + I_{k\ell} \qquad \forall k \in \Gamma(v,i),\ \ell = 1, ..., \bar{L}_v \tag{1.12d}$$

$$\sum_{\ell'=1}^{k} z_{k\ell'} \geq f_k I_{k\ell} \qquad \forall k \in \Gamma(v,i),\ \ell = 1, ..., \bar{L}_v \tag{1.12e}$$

$$b_k, z_{k\ell}, I_{k\ell} \in \{0,1\} \tag{1.12f}$$

The first set of constraints (1.12b) ensure that at most one campaign occupies each slot. The second set of constraints (1.12c) 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 (1.12d) and (1.12e) are the cumulative impression counts $\bar{z}_{k\ell}$. Constraints (1.12d) enforce $I_{k\ell} = 1$ when $\bar{z}_{k\ell} = f_k$, whereas constraints (1.12e) 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 $\psi_{vi}^{*(R)} + \bar{\beta}_{vi}^{*(R)} \geq 0$, 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 1.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 (1.12a).

Computational Experiments:

In all of our computational tests in §1.6, we assumed that each user of visit type v visited the publisher's website a deterministic L_v number of times. This ensured that every pattern was delivered completely to each arriving user. However, in practice the number of visits from each user is never precisely known. 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 §1.5, 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 30 demand nodes and 500 supply nodes. In each supply node, we assumed three user visit-types whose number of visits follows a Poisson distribution at different rates, specified by the vector $\lambda = \{\lambda_1, \lambda_2, \lambda_3\}$. Deterministic pattern lengths, $L = \{L_1, L_2, L_3\}$, employed by our model are fixed[8] 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 1.1). 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 1.1) are chosen to cover at least 99% of the support of the corresponding Poisson distribution (e.g., looking at the first row in Table 1.1, 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

[8]Note that varying L_v parameters will affect $\Gamma(v, i)$, i.e., the connectivity of the graph, and this structural change may affect the under-delivery performance of the deterministic model irrespective of randomness in number of visits from users. Therefore, we did not change L_v parameters and varied the arrival rates.

Visiting Rates (Poisson)	Random Arrival Pattern Lengths	Deterministic Pattern Finish Prob. $(1-\varepsilon)$	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 1.1: Test cases and results under random arrival scenario.
Deterministic pattern lengths are set to $L = \{10, 20, 30\}$ in all cases.

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 the solution is robust to truncation (if a user does not complete the pattern). 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, we moved any empty slots to the ends of patterns, 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 1.1, 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 w. 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 1.1), 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, low-visiting users have an average visit frequency of $\lambda_1 = 17$ while our deterministic and random arrival models use pattern lengths of $L_1 = 10$ (too low) and $\bar{L}_1 = 35$, respectively.

1.F Monolithic Formulation of the Pattern-based R&F Planning Problem

Writing the master problem of Hojjat et al. (2014) in our notation and simplifying yields the following math program, which was used in that paper as a combined reach allocation and pattern assignment step:

$$\text{Minimize} \quad \sum_{k} \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi} w_k}{2\theta_k} \left(x_{vik} - \theta_k\right)^2 + \sum_{k} c_k u_k + \sum_{v,i} \sum_{p\in\mathcal{P}_{vi}} \pi_{vip} y_{vip} \tag{1.13a}$$

$$\text{s.t.} \quad x_{vik} = \frac{1}{s_{vi}} \sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip} \qquad \forall v, i, k \in \Gamma(v,i) \tag{1.13b}$$

$$\sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} + u_k \geq r_k \qquad \forall k \tag{1.13c}$$

$$\sum_{p\in\mathcal{P}_{vi}} y_{vip} \leq s_{vi} \qquad \forall v, i \tag{1.13d}$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \leq 1 \qquad \forall v, i \tag{1.13e}$$

$$0 \leq x_{vik} \leq 1 \tag{1.13f}$$

$$y_{vip} \geq 0,\ u_k \geq 0 \tag{1.13g}$$

The cost parameters π_{vip} explicitly penalize non-smooth and/or non-diverse delivery at the user level in the objective function. Representativeness is redefined at the user level as follows: We would ideally like to spread the reach of campaign k, r_k, uniformly across the eligible number of users $s_k = \sum_{(v,i)\in\Gamma(k)} s_{vi}$; that is, at the rate $\theta_k = r_k/s_k$. Variable y_{vip} denotes the number of time pattern $p \in \mathcal{P}_{vi}$ should be used for type-(v,i) users. The binary indicator parameter b_{kp} denotes whether campaign k is included in pattern p (at the correct frequenct f_k). Constraint (1.13b) captures the variable x_{vik} as a summary statistic of pattern assignment which indicates what proportion of type-(v,i) users are reached by campaign k. This constraint, and variables x_{vik}, can be eliminated by substitution. Constraint (1.13c) is the reach and frequency constraint that replaces the impression-based demand constraint (1.2b). It ensures that campaign k is reached by at least r_k unique users (each viewing the

ad least f_k times) or otherwise u_k counts the shortfall of users. The user-supply constraint (1.13d) ensures that the total number of patterns assigned to (v, i) cannot exceed the number of unique users available (since each user is assigned a single pattern). Constraint (1.13e) is the impression-supply constraint that ensures the total impressions needed to reach a x_{vik} proportion of s_{vi} users by campaigns $k \in \Gamma(v, i)$ (given by $\sum_{k\in\Gamma(v,i)} f_k s_{vi} x_{vik}$) does not exceed the total number of guaranteed impressions available ($L_v s_{vi}$). We can show that this constraint is in fact redundant and dominated by (1.13d), because for any given (v, i):

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

The first equality is given by definition (1.13b). The second equality is a simple rearrangement of terms. The next inequality follows because of the fact that in any pattern it must hold $\sum_{k\in\Gamma(v,i)} f_k b_{kp} \leq L_v$. This is due to fact that we cannot use more than L_v slots in the pattern, and reaching each campaign k occupies f_k slots. The last inequality follows from the user-based supply constraint (1.13d). This shows that the user-based supply constraint is always tighter than the impression-based constraint.

Although x_{vik} represents a proportion, we should point out once more that we do not need constraints of the form $\sum_{k\in\Gamma(v,i)} x_{vik} \leq 1$. This is because a user can be reached by more than one campaign as long as L_v is sufficiently large. Each variable x_{vik} however should be kept between 0 and 1. The upper bound in constraint (1.13f), however, is naturally maintained by constraints (1.13b) and (1.13d).

After eliminating x_{vik} by substitution and removing the redundant constraints (1.13e) and

(1.13e), the formulation simplifies to:

$$\text{(FP):} \qquad \Psi := \text{Minimize} \quad \sum_{k} \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi} w_k}{2\theta_k} \left(\sum_{p\in\mathcal{P}_{vi}} \frac{b_{kp}}{s_{vi}} y_{vip} - \theta_k \right)^2 \qquad \underline{\text{Duals}}(\text{All} \geq 0)$$

$$+ \sum_{k} c_k u_k + \sum_{v,i} \sum_{p\in\mathcal{P}_{vi}} \pi_{vip} y_{vip}$$

$$\sum_{(v,i)\in\Gamma(k)} \sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip} + u_k \geq r_k \qquad \forall k \qquad \alpha_k$$

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

$$y_{vip} \geq 0,\ u_k \geq 0 \qquad \gamma_{win}, \varphi_k$$

In the following section we show the derivation of the column generation subproblem. At a high level, the idea is to start with a small pool of patterns, solve the assignment problem (1.13), 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 collections $\mathcal{P}_{vi}$ and solve the assignment problem again. We repeat this procedure until no improving pattern can be constructed (i.e., full convergence to the optimal solution), or the improvement in the objective function seems negligible. The improvement achieved following each iteration of column generation is not monotonically decreasing; thus, a termination criteria based on absolute or relative improvement in the optimal objective value should be used with caution.

Derivation of Column Generation Subproblem for (FP)

The Lagrangean to 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 with respect to variables y_{vip} gives the reduced cost function:

$$\frac{\partial \mathcal{L}}{\partial y_{vip}} = 0 \quad \Rightarrow \quad \gamma_{vip} \;=\; \sum_{k\in\Gamma(v,i)} \left(\frac{w_k}{\theta_k s_{vi}} \sum_{p\in\mathcal{P}_{vi}} b_{kp} y_{vip} - w_k - \alpha_k \right) b_{kp} + \pi_{vip} + \beta_{vi}$$

An immediate and important observation is that the stationarity condition does not establish a unique relationship between a primal variable, y_{vip}, and the dual variables α_k and β_{vi}. Therefore, the solution obtained from the monolithic math program (FP), unlike the one obtained from (RA) or (RA-δ) is not generalizable. This limitation, greatly diminishes the attractiveness of monolithic modeling approach, i.e., the standard implementation of column generation, in practice.

The column generation subproblem tries to construct a pattern with minimum (negative) reduced cost:

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

Note that we are solving a separate subproblem for each supply node (v,i). These problems can be solved in parallel. If $\psi_{vi}^* + \beta_{vi}^* < 0$ for any supply node (v,i), it is beneficial to add the

pattern to $\mathcal{P}_{vi}$, with $b_{kp} = b^*_k$ and $\pi_{vip} = \pi(\mathbf{b}^*)$, and the solution to (FP) will be improved. Patterns that are unused in the current solution can be deleted for memory efficiency (even though they may return in following iterations). If $\psi^*_{vi} + \beta^*_{vi} \geq 0$ for all (v, i), the solution to (FP) is optimal. The cost function $\pi(\mathbf{b})$ was introduced in Appendix 1.B. To initialize the pattern pools $\mathcal{P}_{vi}$, one can solve subproblems with $\alpha_k = \beta_{vi} = x_{vik} = 0$ (which are primal/dual feasible).

Numerical Experiment

In Hojjat et al. (2014), we tested the monolithic formulation of reach and frequency problem on randomly-generated graphs that we constructed in such a fashion to resemble appropriately-scaled versions of real-world instances. For example, Figure 1.10 demonstrates the progress of the algorithm on a small graph with 40 demand nodes and 300 supply nodes. Each supply node was further partitioned into 3 visit types with guaranteed visit lengths of $\{10, 20, 30\}$ impressions. There were approximately 4600 arcs in the graph (40% of the total possible connections). The horizontal axis shows time (in seconds). Each vertical dashed line shows an epoch where the master problem is solved, and the thick black curve tracks the optimal value of the master problem, denoted Ψ^*. In between the epochs, we solved the subproblems until (at most) 20 improving patterns were found. The red curves show the cumulative number of new patterns found during each epoch, and the green curve shows the total number of patterns existing in the master problem. Throughout the process, we deleted old unused patterns to keep the total number of available patterns at each point in time from growing too quickly. We solved the subproblems in an ad-hoc (essentially random) supply node order. We used only a diversity-seeking metric for pattern quality. We used the AMPL modeling language with CPLEX solver on a dual core i5 2.5GHz CPU with 8GB of RAM to carry out the experiment.

The master problem fully converged to the optimal solution after 10 iterations (6 minutes),

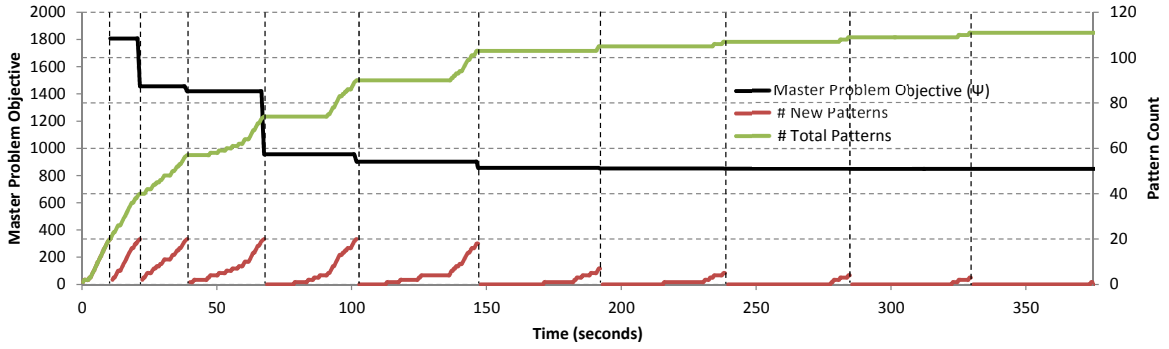


Figure 1.10: Performance of the Monolithic CG Model on a Sample Graph
(40 contracts, 300 supply pools, 3 user visiting types, 40% graph connectivity)

at which point we solved all $300 \times 3 = 900$ subproblems to verify that no improving pattern existed. As we can see in Figure 1.10, the optimal value Ψ^* initially improves quickly, but the rate of improvement tapers off, becoming negligible beyond iteration 6 (2.5 minutes). Note that the subproblems are not being solved in parallel in our numerical experiment. With full parallelization, the full convergence could be attained in less than 1 minute. Moreover, there is a tradeoff between the number of iterations it takes for the master problem to converge and the maximum number of new patterns we aim for during each epoch. With no limit on the number of new patterns, the above example would converge in 4 iterations; however, 900 subproblems need to be solved in each iteration, and the total run time happens to be worse than 6 minutes.

Note that among the possible $O\left(10^{19}\right)$ patterns that can be constructed for this small instance[9], only 111 are used in the final solution.

Finally, we would like to point out that the improvement in the optimal value of the master problem is not guaranteed to be monotonically decreasing. For instance, the improvement in Ψ^* in iterations 3 and 5 was very low, whereas a number of patterns were found during iteration 4 which drastically improved the solution. Therefore, a termination criteria based

[9]If we differentiate patterns based on the exact arrangement of ads within the pattern, we can construct $\sum_L 40^L = 1.15 \times 10^{48}$ patterns, given $L \in \{10, 20, 30\}$. If we differentiate only based on the number of times each campaign appears in the pattern, we can construct $\sum_L \sum_{c=1}^{L} \binom{40}{c} \binom{L-1}{c-1} \approx 31.63 \times 10^{18}$ patterns.

on the absolute or relative improvement in Ψ^* should be used with great caution.

1.G 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:

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

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 $\alpha_k^* = 0$:

$$\begin{aligned}
\sum_{(v,i)\in\Gamma(k)} s_{vi}x_{vik}^* &= \sum_{(v,i)\in\Gamma(k)} s_{vi}g_{vik}(0,\beta_{vi}^*) \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\min\left[1,\max\left[0,\theta_k\left(1-\frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right)\right]\right] \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\max\left[0,\theta_k\left(1-\frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right)\right] \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k\max\left[0,\ 1-\frac{1}{w_k}\frac{f_k}{L_v}\beta_{vi}^*\right] \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k \quad = \quad r_k. \qquad (1.14)
\end{aligned}$$

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

$$\text{(RA-}\delta\text{)} \quad \equiv \quad \text{Minimize} \quad \sum_k \sum_{(v,i)\in\Gamma(k)} \frac{s_{vi}}{2\theta_k} w_k \left(x_{vik} - \theta_k\right)^2 + \sum_k c_k \big(r_k - \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik}\big) \tag{1.15a}$$

$$\text{s.t.} \quad \sum_{(v,i)\in\Gamma(k)} s_{vi} x_{vik} \le r_k \qquad \forall k \tag{1.15b}$$

$$\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} x_{vik} \le \delta_{vi} \qquad \forall v, i \tag{1.15c}$$

$$0 \le x_{vik} \le 1 \qquad \forall v, i, k \in \Gamma(v,i) \tag{1.15d}$$

The constraint (1.15b) is corresponding to $u_k \ge 0$. 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. □

1.H Derivation of the Dual Problem for (RA-δ):

The Lagrangean dual function is given by: $\mathcal{L}(\alpha, \beta, \gamma, \varphi) = \min_{x,u} \mathcal{L}(x, u; \alpha, \beta, \gamma, \varphi)$.

Substituting $x^*_{vik} = \theta_k + \frac{\theta_k}{s_{vi} w_k}\big(s_{vi}\alpha_k - \frac{f_k}{L_v} s_{vi}\beta_{vi} + \gamma^L_{vik} - \gamma^U_{vik}\big)$ from stationarity condition (ST1) from previous section into the Lagrangean function $\mathcal{L}(x, u; \alpha, \beta, \gamma, \varphi)$, and realizing that the coefficient of u_k in the Lagrangean function is zero per (ST2), i.e., $c_k - \alpha_k - \varphi_k = 0$, we can write the Lagrangean dual function as:

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

The dual program to (RA-δ) is therefore a convex quadratic program with non-negative

variables and linear constraints:

$$\begin{aligned} \text{(RA-}\delta\text{D):} \quad \text{Maximize} \quad & \mathcal{L}(\alpha, \beta, \gamma, \varphi) \\ \text{s.t.} \quad & \alpha_k + \varphi_k = c_k \qquad \forall k \\ & \alpha_k,\ \beta_{vi},\ \gamma^U_{vik},\ \gamma^L_{vik},\ \varphi_k \ \geq 0 \end{aligned}$$

1.I 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. The idea is to show 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. We then show that the resulting solution satisfies all KKT conditions and since the problem (RA-δ) is convex, the obtained solution must be optimal.

Convergence:

Let α_k^t and β_{vi}^t denote the dual values computed in iteration t of SHALE. 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:

$$\begin{aligned}
\Big|r_k(\alpha_k^t,\boldsymbol{\beta}^t)-r_k(\alpha_k^{t-1},\boldsymbol{\beta}^t)\Big| &= \left|\sum_{(v,i)\in\Gamma(k)} s_{vi}g_{vik}(\alpha_k^t,\beta_{vi}^t) - s_{vi}g_{vik}(\alpha_k^t,\beta_{vi}^t)\right| \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|g_{vik}(\alpha_k^t,\beta_{vi}^t) - g_{vik}(\alpha_k^t,\beta_{vi}^t)\right| \\
&= \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|\ \mathrm{sat}\Big[0,1,\theta_k+\frac{\theta_k}{w_k}\Big(\alpha_k^t-\frac{f_k}{L_v}\beta_{vi}^t\Big)\Big] - \mathrm{sat}\Big[0,1,\theta_k+\frac{\theta_k}{w_k}\Big(\alpha_k^{t-1}-\frac{f_k}{L_v}\beta_{vi}^t\Big)\Big]\right| \\
&\leq \sum_{(v,i)\in\Gamma(k)} s_{vi}\left|\frac{\theta_k}{w_k}\Big(\alpha_k^t-\alpha_k^{t-1}\Big)\right| \\
&= \frac{r_k}{w_k}\Big|\alpha_k^t-\alpha_k^{t-1}\Big| \qquad (1.16)
\end{aligned}$$

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, $\big|\min\big[1,\max[0,a]\big]-\min\big[1,\max[0,b]\big]\big| \leq |a-b|$. (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 (1.16) suggests:

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

$$\Rightarrow \qquad \alpha_k^{t_0} \;\geq\; \alpha_k^{t_0-1} + w_k\Big(1 - \frac{r_k(\alpha_k^{t_0-1}, \boldsymbol{\beta}^{t_0})}{r_k}\Big) \;\geq\; \alpha_k^{t_0-1} \tag{1.17}$$

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 (1.16) 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 (1.17) 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:

$$r_k(\alpha_k^{t_0}, \boldsymbol{\beta}^{t_0+1}) \;=\; \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0+1}) \;\leq\; \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(\alpha_k^{t_0}, \beta_{vi}^{t_0}) \;=\; r_k(\alpha_k^{t_0}, \boldsymbol{\beta}^{t_0}) \;\leq\; r_k \quad (1.18)$$

which is the Condition 1 for Iteration $t_0 + 1$. This implies that all $\alpha_k^{t_0+1} \geq \alpha_k^{t_0}$ in Step-2 of iteration $t_0 + 1$, per (1.17), and therefore all $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ values will monotonically increase in all iterations $t \geq t_0$, and Condition 1 will be maintained throughout. Since α_k is bounded above by c_k, the algorithm must converge.

Intuitively, Condition 1 requires that no campaign is over-delivered. Then we could imagine that in each α_k update, we seek to eliminate under-delivery for 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 violate the supply constraint for 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 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 pass of $\boldsymbol{\beta}$ updates has been performed to correct complementary slackness, then we never need to decrease β_{vi} values as they will take their lower-bound of 0 when 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 $r_k(\alpha_k^0, \boldsymbol{\beta}^1) \leq \sum_{(v,i)\in\Gamma(k)} s_{vi} g_{vik}(0,0) = \sum_{(v,i)\in\Gamma(k)} s_{vi}\theta_k = r_k$. 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, $r_k(\alpha_k^*(\delta'), \boldsymbol{\beta}^*(\delta')) \leq r_k$ (see (1.14) in Appendix 1.G that shows over-delivery never occurs in the optimal solution). Assume we need to solve a new instance (RA-δ) in which $\delta_{vi} \leq \delta'_{vi}$ for all (v,i). Initializing $\alpha_k^0 = \alpha_k^*(\delta')$, note that if at any node (v,i) we happen to have $\sum_{k\in\Gamma(v,i)} \frac{f_k}{L_v} g_{vik}(\alpha_k^0, \beta_{vi}^*(\delta')) \leq \delta_{vi} \leq \delta'_{vi}$, 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 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 $\beta_{vi}^1 \geq \beta_{vi}^*(\delta)$ for every (v,i). From (1.18) we obtain that $r_k(\alpha_k^0, \beta^1) \leq r_k(\alpha_k^*(\delta), \beta^*(\delta)) \leq r_k$ 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.

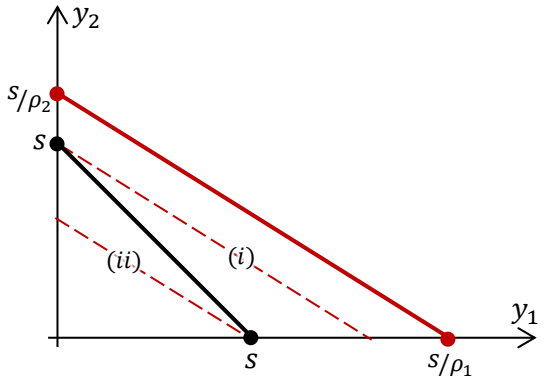
As a part of proving the convergence of the algorithm, we showed that no campaign will experience over-delivery in any iteration subsequent to meeting Condition 1. We also showed that the primal solution always satisfies the supply constraints after the Step-1 $\boldsymbol{\beta}$ updates. So, after 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:

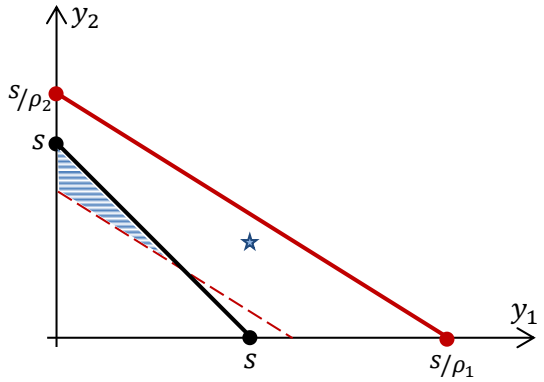
The optimality bound, due to Bharadwaj et al. (2012), is based on the argument that for any $t \geq t_0$, if for some k with $\alpha_k^t \neq c_k$ we have $r_k(\alpha_k^{t-1}, \beta^t) \leq (1-\varepsilon)r_k$, then (1.17) implies $\alpha_k^t \geq \alpha_k^{t_0-1} + w_k\varepsilon$. That is, α_k increases 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.

1.J Geometric Illustration of δ Updates

In solving the (PA) we take the approach of relaxing the user-supply constraint (1.4c) so a feasible solution is guaranteed and easy to construct to initialize our column generation procedure. However, note that the constraint set (1.4b) together with the impression-supply



(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 1.11: 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 $\delta = 1$.

constraints (1.3c) 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 (1.5b)). Figure 1.11 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 $\delta^{\min}_{vi}$ and $\delta^{\max}_{vi}$ respectively denote the minimum (non-empty) and maximum impression utilization rates possible for supply node (v,i). Obviously, $\delta^{\min}_{vi} = \min_{k\in\Gamma(v,i)}\{f_k\}/L_v$, i.e., the pattern consisting of only the campaign with smallest f_k; and $\delta^{\max}_{vi}$ 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 1.11(a) illustrate the implied constraint $\sum_{p \in \mathcal{P}_{vi}} \rho_{vip} y_{vip} \leq s_{vi} \delta_{vi}$ when δ_{vi} is exactly at $\delta_{vi}^{\min}$ or $\delta_{vi}^{\max}$.

In the absence of the user-supply constraint (1.4c), i.e., the solid black line, our approach is to adjust the δ_{vi} values until the implied constraints $\sum_{p \in \mathcal{P}_{vi}} \rho_{vip} y_{vip} \leq s_{vi} \delta_{vi}$ push the optimal solution of (PA) to satisfy $\sum_{p \in \mathcal{P}_{vi}} y_{vip} \leq s_{vi}$. Considering the slope differences between these two types of constraints, Figure 1.11(b) shows that, a certain portion of the feasible region (hatched in blue) may be cut off. This causes the solution produced by our hierarchical formulation to be suboptimal to that of our monolithic formulation (1.13). 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. Our numerical tests on realistic instances that match industry data reveal that the degree of suboptimality of our hierarchical formulation relative to our monolithic formulation is within 1-3 percent, depending on the instance. Overall, we feel this is reasonable given the other advantages that our hierarchical formulation has over our monolithic formulation as described in §1.4.

Moreover, we note that in §1.5.4 we adopted the simplest update rule for δ values, and that more advanced update rules may reduce this gap. For instance, we noticed if we update only a fraction (and not all) of the δ_{vi} values in each iteration (especially, if chosen based on the smallest β_{vi} value, i.e., the least impact on the objective of (RA-δ)), the optimality gap can be further reduced in most instances.

1.K Equivalence of Scrap- and Roll-minimizing Cutting stock Problems

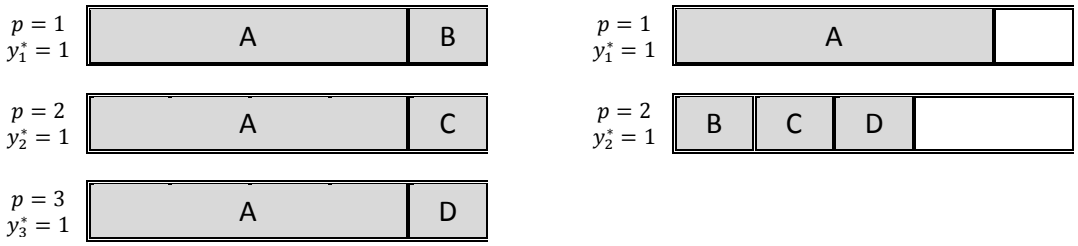
Consider the classic cutting stock problem where a manufacturer has an infinite stock of metal rolls (or rods) of fixed length L, and there is a demand r_k for pieces of length $f_k < L$. The manufacturer may minimize scrap (pieces of roll that are not of usable length and must be scrapped) by generating 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 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:

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

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 1.12 shows that the scrap-minimizing



(a) Waste-Minimizing Solution

(b) Roll-minimizing Solution

Figure 1.12: Comparison of optimal solutions to a cutting stock problem when demand constraints are expressed as inequalities (i.e., over-production is allowed)

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.

CHAPTER 2:

Controlling the Exposure Frequency Distribution of Online Advertising with Markov Chains

2.1 Introduction

Online advertising has become the largest section of the advertising market with an annual revenue of $49.5 billion (IAB 2016). Online platforms are interactive, easily customizable, and provide world-wide reach and great control over targeting. Latest industry trends show that *people-based marketing* has become a particularly popular catchphrase in the online advertising industry and major efforts are being exerted by online publishers to track individuals (Kattula et al. 2015). In the meantime, the exponential growth in the use of portable devices 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 and across multiple devices. However, the existing forms of advertising campaigns, and respectively, the ad serving mechanisms employed by online publishers, do not utilize the capabilities of the online platform to the fullest potential. In particular, most existing forms of online ad campaigns simply 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. While metrics of reach (how many unique individuals are exposed to the ad) and frequency (how many times each individual is exposed to the ad) have been an indispensable part of marketing literature and always an integral part of designing and measuring classical advertising campaigns, they are not being incorporated into how online ad campaigns are

contractually defined. There is evidence that online advertisers are becoming more concerned about who they reach (Warc 2015) and reach and frequency metrics such as Gross Rating Points (GRP) are on demand by online advertisers (eMarketer 2009). Furthermore, recent studies show that breaking the advertising message into multiple bite-size pieces along with a creative sequencing of those pieces (e.g., to convey a story about the product or service) could drastically increase click-through and subscription rates (Adaptly 2014). Therefore, there is great benefit in having an ad campaign which contractually obligates the publishers to provide a specific reach and frequency.

In this paper, we introduce and study a new form of reach and frequency (R&F) contract in which an advertiser specifies the fraction of users from a desired target audience she would like to reach (by at least 1 impression) as well as a frequency distribution which specifies what proportion of users should see the ad how many times. For instance, the advertiser might want to reach 1000 unique individuals, and in particular, want 25% to be exposed twice, 50% to see the ad 3 times, and 25% to see it 4 times. This is a generalization of the first type of R&F contract proposed by Hojjat et al. (2014) in which the frequency is defined as a single exposure count that every individual is supposed to meet in order to be considered as reached. In fact, the R&F contract proposed by Hojjat et al. (2014) is equivalent to a frequency distribution in which the mass of audience the advertiser wishes to reach is placed entirely on a single frequency number.

The number of visits made by each user throughout a campaign's horizon is not deterministic. Therefore, it is a challenge for the publisher to deliver reach and frequency as requested by the advertiser. Upon each user visit, the publisher has a split-second time to decide whether or not an impression of the ad should be shown. This paper shows that we can always either solve the publisher's decision problem and determine the allocation rule, or show that the advertiser's reach and frequency specification is infeasible to implement for the arrival process to publisher's website.

Our manuscript is organized as follows: In §2.2 we provide an overview of relevant literature. In §2.3 we solve the publisher's problem assuming that the frequencies are counted throughout the campaign's horizon, or non-overlapping sub-intervals of the horizon of certain length (e.g., each calendar week). In §2.3.1 we develop a Markov chain model for this problem. §2.3.2 is devoted to characterizing the assignment rule for the publisher. We develope a math programming formulation (§2.3.2.1) and then discuss a simplified Markov chain model (§2.3.2.2) which can be solved in closed form and helps us characterize the necessary and sufficient condition for feasibility. We present a numerical example in §2.3.3 and then consider extensions of our basic model in §2.3.4. In particular, we discuss how our model can be generalized to plan multiple ad campaigns (§2.3.4.1), and how the publisher may distribute the frequency distribution among multiple classes of users with different visit patterns (§2.3.4.2).

One of our findings with fixed-horizon R&F contracts is that simplistic solutions which are easy to obtain may result in poor spreading of campaigns over time. To alleviate this issue, in §2.4, we consider an alternative interpretation of our R&F contract in which frequencies are counted on a rolling basis. For example, the advertiser may monitor the number of exposures delivered to individuals by sampling arbitrarily-chosen 24-hour periods throughout the campaign's horizon and expect each user to have seen the ad $\{0, 1, 2, 3\}$ times with a sample probability of $\{5, 10, 25, 60\}\%$, respectively. For this variant of the R&F contract, we develop a birth-and-death Markov chain model for this problem in discrete (§2.4.1) and continuous (§2.4.2) time, and for each model we derive the exposure rates publisher should use, and characterize the feasibility conditions. Illustrative examples for this type of contract are provided in §2.4.3. It is observed that when the advertiser's specified frequency distribution has a low variance, the rolling horizon approach naturally attains uniform spread of campaigns over their horizon.

For each of the fixed- and rolling-horizon models we consider in this paper, we find that the feasibility criteria are easy to check, and once those criteria are met, obtaining the publisher's

assignment rule can be done very quickly and efficiently in linear time in the length of the frequency distribution specified by the advertiser. We believe our modeling and solution approach can be useful in practice, and certainly help toward a deeper understanding of serving reach and frequency contracts in online media. Our concluding remarks and directions for future work appear in §2.5.

2.2 Literature Review

The marketing literature on the impact of ad repetition on user brand recall and conversion is quite rich. These studies commonly agree that the marginal benefit of additional exposures is increasing at initial exposures (*wearin* effect) and starts to decline beyond some threshold (*wearout* effect) where too much exposure creates tedium/boredom (Campbell and Keller 2003; Yaveroglu and Donthu 2008). Therefore, there exists an optimal level of exposure that maximizes advertising effectiveness and conversion rate. Chandler-Pepelnjak and Song (2003) show how the most efficient or most profitable frequency rates for an online campaign can be determined from historical performance. In general, one can classify most ad campaign in two groups: Those that aim for brand awareness, and those announcing a specific promotion. The former typically requires high reach and low frequency, whereas the latter is expected to require higher frequency to induce a conversion/purchase with little attention to reach (as long as a desired number of conversions is attained). Our model does not recommend appropriate reach and frequency levels for advertisers. Instead, we take them as parameters and recommend an impression allocation rule for the publisher to meet these reach and frequency requirements when user visits are non-deterministic.

The problem of optimally allocating the supply of impressions to advertising campaigns has also been studies quite extensively in the literature using a variety of modeling and solution techniques, such as Linear Programming (Chickering and Heckerman 2003; Nakamura and Abe 2005), Quadratic Programming (Turner 2012; Bharadwaj et al. 2012), competitive

primal-dual algorithms (Mehta 2012), queuing theory (Najafi Asadolahi and Fridgeirsdottir 2014), and revenue optimization (Roels and Fridgeirsdottir 2009; Balseiro et al. 2014).

Hojjat et al. (2014) was the first to introduce reach and frequency contracts and develop an optimization framework for simultaneous planning of R&F campaigns. In their definition of R&F contract, each campaign defines a required number of times (i.e., the frequency) that each individual should be exposed to the ad for him/her to be considered as reached. This implies that the publisher receives no revenue for any impression delivered in excess of the specified frequency, or all impressions delivered to a user if their total count does not meet the frequency requirement. Several issues can be pointed out with regards to this notion of R&F. First, the research shows that all exposures to display ads, regardless of whether they have been clicked on or even directly looked at, do get processed by users at a pre-attentive level which does impact brand awareness (Briggs and Hollis, 1997; Dreze and Hussherr, 2003). Therefore, impressions of ads in less/excess of the most efficient/profitable frequency should still be of some value to advertisers. Second, and as evidenced in the computational studies of Hojjat et al. (2016), a major challenge in planning such R&F campaigns is to maintain *waste*, i.e., impressions which are served but not billable because the failed to meet or exceeded the target frequency requirements, at minimum. Waste is a natural consequence of randomness in the number of visits made by each user throughout the campaign's horizon. For example, suppose an advertiser wishes to reach every individual within a certain demographic at a frequency of 3. This implies that the publisher must serve all users with 3 impressions of this campaign. If only 70% of users make more than 2 visits, then 30% of the target audience can never be reached by this campaign and any impression served to them upon the first and second visits will be wasted. Given that waste is virtually unavoidable and the publisher is only paid by the number of individuals reached at the correct frequency, the advertiser never observes the *true* frequency distribution which was served in the target audience. As well, the publisher's estimate of the average waste would appear as a hidden cost in the pricing of R&F

campaign. We believe our new definition of R&F contract eliminates this untruthfulness and allows the publisher to be paid for every impression served, as long as the aggregate frequency distribution meets the advertiser's specification.

There are existing models that cast the publisher's revenue optimization as a Markov decision process (e.g., Archak et al. 2010; Truzzi et al. 2012). In these models, ad impressions must be served to a stochastic stream of arrivals, one at a time. Each impression contributes a revenue (or consumes certain budget) and the objective is to maximize the expected total revenue throughout the planning period. Our modeling approach using Markov chains, however, is quite different and can be considered new in the context of online advertising. We do not model revenue explicitly, but aim to implement advertiser's reach and frequency requests perfectly, in expectation. Therefore, our decisions (i.e., ad exposure rates) do not contribute a direct reward to an objective function. Instead, they influence the transition probabilities of a user's state (i.e., the current frequency count). The goal is to find a set of exposure rates such that the *steady-state distribution* of the Markov chain matches the frequency distribution specified by the advertiser. Therefore, we contribute to the literature on the *inverse problem of Markov chain* which surprisingly has not received much attention. Existing literature (e.g., Ray and Margo 1976; Morimura et al. 2013) is fairly limited and does not generalize to our particular application.

2.3 Fixed-Horizon Frequency Specification

In our first modeling approach, we assume that a single advertiser wishes to achieve the frequency distribution $\boldsymbol{\pi} = (\pi_0, \pi_1, ..., \pi_F)$, where π_k denotes the proportion of users that see their ad k times over a fixed planning period. The planning period could be the campaign's full horizon H, or we could consider subdividing the full horizon into a number of non-overlapping time intervals of length T (e.g., a calendar week). In the latter case, we assume that all frequency counts are reset to zero at the start of each T interval. The distribution

$\boldsymbol{\pi}$ also implies that the advertiser wishes to reach a $\theta = 1 - \pi_0 = \sum_{k=1}^{F} \pi_k$ proportion of users at least once. In practice, it is more likely that the advertiser would directly specify the reach fraction, θ, along with a frequency distribution $\boldsymbol{\pi'} = (\pi'_1, ..., \pi'_F)$ which applies only to users who are to be reached. In this case, the advertiser's specifications translate into $\pi_0 = 1 - \theta$ and $\pi_k = \theta\pi'_k$ for $k = 1, \ldots, F$ in our model. Note that specifying reach as a percentage of audience size is a common practice and such campaigns are known as *share-of-voice* campaigns. Gross Rating Points (GRPs), defined as percent reach multiplied by average frequency, are arguably the most common way that existing campaigns are defined/measured which also use reach as a proportion of audience size. We use $\Pi_k = \sum_{k'=0}^{k} \pi_{k'}$ to represent the cumulative distribution of $\boldsymbol{\pi}$.

We further assume that the number of page visits from users throughout the planning period are independent and follow the same probability distribution $\boldsymbol{\phi} = (\phi_0, \phi_1, ..., \phi_L)$. That is, each user makes exactly t visits with probability ϕ_t. We use $\Phi_t = \sum_{t'=0}^{t} \phi_{t'}$ to denote the cumulative distribution of $\boldsymbol{\phi}$. Later in §2.3.4 we demonstrate how our model can be extended to the case of multiple user classes with different visit probability distributions. Without loss of generality, we assume $L \geq F$, i.e., the maximum number of visits expected from a user is no less than the maximum frequency requested; otherwise, the problem is trivially infeasible. Note that in practice, $\boldsymbol{\phi}$ would be derived from historical visit patterns. Therefore, it is practical to assume that a non-zero (ϕ_0) proportion of identified users will not make a visit throughout a planning period in the future.

Upon each user visit, the publisher needs to make a decision as to whether or not he should expose the user to the ad, given the number of times the user has visited the website so far, how many times she has already been exposed to the ad, and the expected probability of future visits. In this section we characterize the feasibility criteria of such a decision and the publisher's decision rule when those feasibility criteria are met.

2.3.1 Markov Chain Model

Upon the t'th visit from a user, we denote the user's state by a tuple (t, k) where $k < t$ is the number of times the user has been exposed to the ad over the past $t-1$ visits. A user who has seen the ad k times but will never make another visit to the publisher's website is defined to be in an absorbent state $[k]$.

Let $x_{t,k}$ denote the probability that the publisher shows the ad to a user in state (t, k). Given that the user has made t visits, there is a $\phi_t/(1-\Phi_{t-1})$ probability that this t'th visit will be her last and the user will never visit again. Therefore, with probability $x_{t,k}\phi_t/(1-\Phi_{t-1})$ the user will be absorbed in state $[k+1]$ and with probability $(1-x_{t,k})\phi_t/(1-\Phi_{t-1})$ she will move to the absorbent state $[k]$. Similarly, there is a $(1-\Phi_t)/(1-\Phi_{t-1})$ probability that the user will make at least one more visit. Therefore, with probability $x_{t,k}(1-\Phi_t)/(1-\Phi_{t-1})$ the user revisits in state $(t+1, k+1)$, and with probability $(1-x_{t,k})(1-\Phi_t)/(1-\Phi_{t-1})$ she returns in state $(t+1, k)$. The transition diagram and probabilities are depicted in Figure 2.1. Absorbent states are depicted as vertical bars. Note that all users start from state $(0, 0)$ where they have never arrived and have not seen any ads yet. As we can see, a ϕ_0-proportion of users never arrive throughout the planning period. The remaining $(1-\phi_0)$ fraction make their first visit in state $(1, 0)$ at which point the publisher gets to make an assignment decision.

2.3.2 Derivation of Exposure Probabilities

The publisher's problem is to find the exposure rates $x_{t,k}$ for each $k \in \{0, \ldots, F\}$ and $t \in \{k+1, \ldots, L\}$ such that the total fraction of traffic absorbed in each state $[k]$ equals π_k. The following theorem provides a necessary condition for feasibility:

Theorem 3 (Necessary Condition for Feasibility)**.** *For the publisher to be able to implement the frequency distribution $\boldsymbol{\pi} = (\pi_0, \pi_1, ..., \pi_F)$, it must be that $\Pi_k \geq \Phi_k$ for every $k \in \{0, ..., F\}$. That is, the number of user visits must first-order stochastically dominate the advertiser's*

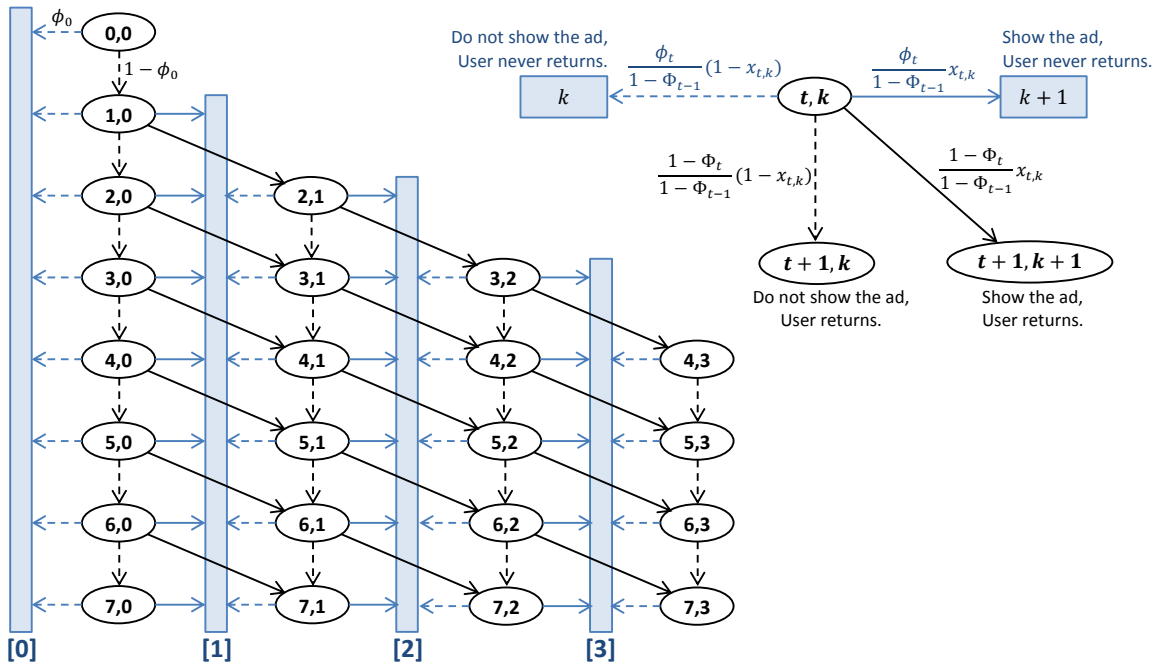


Figure 2.1: States and Transition Probabilities in the Fixed-Horizon Markov Chain Model (Assuming $F = 3$ and $V = 7$)

requested frequency.

Proof. For a user to see the ad k times, she should make at least k visits. For every $k \in \{1, ..., F\}$, the proportion of users who should be exposed to the ad at least k times (i.e., $1 - \Pi_{k-1}$) cannot exceed the proportion of users who make k or more visits (i.e., $1 - \Phi_{k-1}$). Therefore, we must have $1 - \Pi_{k-1} \leq 1 - \Phi_{k-1}$, or $\Pi_{k-1} \geq \Phi_{k-1}$. □

We now explore two approaches in solving this problem. First we show that the problem can be cast as a network flow problem. Then we derive a closed-form solution for a restricted version of the problem which results in a simplified Markov Chain. We use this closed-form solution to characterize the sufficient condition for feasibility.

2.3.2.1 Solving for Exposure Rates

The problem of finding exposure rates $x_{t,k}$ to direct a π_k-fraction of users to each absorbent state $[k]$ involves solving a system of equations which result from the balance equations of

the Markov chain model. Let $f_{S1,S2}$ denote the fraction of traffic from state $S1$ directed to state $S2$. For instance, $f_{(4,1),[2]}$ is the fraction of traffic at state $(4,1)$ that is directed to the absorbent state [2]. We know from the transition probabilities of the Markov chain that this flow corresponds to a $x_{4,1}\phi_4/(1-\Phi_3)$ fraction of the input flow into state $(4,1)$ which is $f_{(3,0),(4,1)} + f_{(3,1),(4,1)}$. Therefore, we can compute the exposure rate as $x_{4,1} = (1-\Phi_3)f_{(4,1),[2]}/\big(\phi_4(f_{(3,0),(4,1)} + f_{(3,1),(4,1)})\big)$. In general, once all the flows $f_{S1,S2}$ are found, the exposure probabilities are given by:

$$x_{t,k} \;=\; \frac{1-\Phi_{t-1}}{\phi_t}\,\frac{f_{(t,k),[k+1]}}{f_{(t,k),[k]} + f_{(t,k),[k+1]} + f_{(t,k),(t+1,k)} + f_{(t,k),(t+1,k+1)}} \qquad \begin{array}{l}\forall k \in \{1,...,F\},\\ \forall t \in \{k+1,...,L\}\end{array} \tag{2.1}$$

The flows $f_{S1,S2}$ must satisfy the following flow balance constraints which ensure that the outflow from each node equals the inflow into that node:

$$f_{(0,0),[0]} = \phi_0, \qquad f_{(0,0),(1,0)} = 1-\phi_0 \tag{2.2}$$

$$\forall t \in \{1,...,T\}: \quad f_{(t,0),[0]} + f_{(t,0),[1]} + f_{(t,0),(t+1,0)} + f_{(t,0),(t+1,1)} \;=\; f_{(t-1,0),(t,0)} \tag{2.3}$$

$$\forall k \in \{1,...,F\}: \quad f_{(k+1,k),[k]} + f_{(k+1,k),[k+1]} + f_{(k+1,k),(t+1,k)} + f_{(k+1,k),(t+1,k+1)} \;=\; f_{(k,k-1),(t,k)} \tag{2.4}$$

$$\begin{array}{r}\forall k \in \{1,...,F\}\\ \forall t \in \{k+2,...,L\}\end{array}: \quad \begin{array}{l} f_{(t,k),[k]} + f_{(t,k),[k+1]} + f_{(t,k),(t+1,k)} + f_{(t,k),(t+1,k+1)}\\ \qquad\qquad = \; f_{(t-1,k),(t,k)} + f_{(t-1,k-1),(t,k)}\end{array} \tag{2.5}$$

Furthermore, note that the publisher has only one degree of freedom at every node in splitting the flow (i.e., the exposure rate $x_{t,k}$). The outgoing flows from every node (t,k) with $k \in \{0,...,F\}$ and $t \in \{k+1,...,L\}$ must satisfy the following relationships:

$$f_{(t,k),[k]} \;=\; \frac{\phi_t}{1-\Phi_t} f_{(t,k),(t+1,k)} \tag{2.6}$$

$$f_{(t,k),[k+1]} \;=\; \frac{\phi_t}{1-\Phi_t} f_{(t,k),(t+1,k+1)} \tag{2.7}$$

$$f_{(t,k),[k]} + f_{(t,k),[k+1]} \;=\; \frac{\phi_t}{1-\Phi_{t-1}}\Big(f_{(t,k),[k]} + f_{(t,k),[k+1]} + f_{(t,k),(t+1,k)} + f_{(t,k),(t+1,k+1)}\Big) \tag{2.8}$$

A total flow of π_k should be directed to each absorbent state $[k]$, that is:

$$k = 0: \quad \sum_{t=0}^{L} f_{(t,0),[0]} \;=\; \pi_0 \tag{2.9}$$

$$\forall k \in \{1, ..., F\}: \quad \sum_{t=k}^{L} f_{(t,k-1),[k]} + \sum_{t=k+1}^{L} f_{(t,k),[k]} \;=\; \pi_k \tag{2.10}$$

and finally, all flows $f_{S1,S2}$ must be non-negative:

$$f_{S1,S2} \geq 0 \qquad \begin{array}{c} \forall k \in \{0, ..., F\},\ t \in \{k+1, ..., L\} \\ S_1 \in (t,k),\ S_2 \in \{(t+1,k) \cup (t+1,k+1) \cup [k] \cup [k+1]\}. \end{array} \tag{2.11}$$

One could employ a variety of standard techniques to find a feasible solution to (or detect infeasibility of) the above set of linear equations and inequalities (2.2)–(2.11). For instance, one could use an LP-solver.

We should point out that when the problem is feasible, it is often the case that the solution is not unique. By using a quadratic objective which minimizes the variance of flows across all links in the network (e.g., minimizing the sum of squared flow values) along with constraints (2.2)–(2.11), the publisher can produce a more well-spread solution.

In the case of infeasibility, the publisher can relax constraints (2.9) and (2.10) and invoke a Quadratic Programming (QP) solver to find a solution which minimizes the deviations from the desired distribution as follows:

$$\text{Minimize}: \quad \Big(\sum_{t=0}^{L} f_{(t,0),[0]} - \pi_0\Big)^2 \;+\; \sum_{k=1}^{F} \Big(\sum_{t=k}^{L} f_{(t,k-1),[k]} + \sum_{t=k+1}^{L} f_{(t,k),[k]} - \pi_k\Big)^2 \tag{2.12}$$

This will attain a well-spread solution which is as close to the advertiser's ideal $\boldsymbol{\pi}$ as possible.

2.3.2.2 Simplified Markov Chain for the Restricted Problem

Consider a modified version of the problem as depicted in Figure 2.2. This is a restriction of the original problem from Figure 2.1 in which all exposure rates $x_{t,k}$ for $t \geq k+2$ are restricted to be zero. Essentially, upon the user's k'th visit, we decide on the probability of

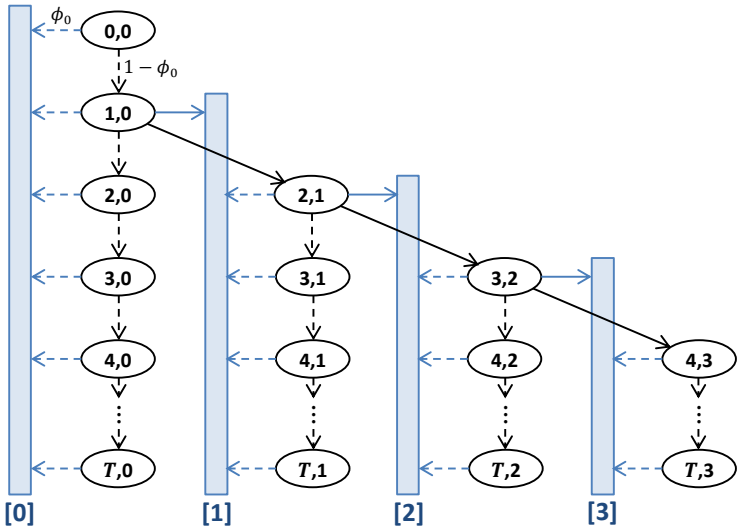


Figure 2.2: Restricted Markov Chain for the Fixed-Horizon Model

continuing to show an ad to the user. If so, the user either transitions from state $(k, k-1)$ to $(k+1, k)$ upon the subsequent visit or she might be be absorbed in state $[k]$ with no future visits. On the other hand, if we choose not to show the ad to the user, we commit to not showing the ad to that user upon any subsequent visit. In other words, $1 - x_{k,k-1}$ gives the stopping probability for ad exposure upon k'th arrival.

Theorem 4 (Feasibility Equivalence)**.** *If the original problem has a feasible solution, then so does the restricted problem, and vice versa.*

Proof. Assume that a feasible solution to the original problem (Figure 2.1) is given. Focusing on all states $(t, 0)$, we must have $\sum_{t=1}^{T} f_{(t,1),[0]} = \pi_0 - \phi_0$ (i.e., a total flow of π_0 is directed to $[0]$) and inevitably $\sum_{t=1}^{T} f_{(t,0),[1]} + f_{(t,0),(t+1,1)} = 1 - \pi_0$ (the rest is directed toward $[1]$ and $(t, 1)$ states). Due to flow conservation, we have $(\pi_0 - \phi_0) + (1 - \pi_0) = 1 - \phi_0$, i.e., total outflow from $(1, 0)$ equals the inflow. Therefore, there must exist a rationing of the flow $1-\phi_0$ in the restricted problem (Figure 2.2) such that $\pi_0 - \phi_0$ is directed toward $[0]$ and the rest, inevitably $1 - \pi_0$, is directed toward $[1]$ and $(2, 1)$. Now, for any $k \geq 1$: the total inflow into $[k]$ and states $(t, k), \forall t > k$ in the existing feasible solution must be $1 - \Pi_{k-1}$, and the total

outflow from states (t,k) on to $[k+1]$ and $(t+1,k+1)$ must be $1-\Pi_k$. If the same amount of flow $(1-\Pi_{k-1})$ is transferred in two streams instead (as in the restricted problem), i.e., from $(k,k-1)$ to $[k]$ and $(k+1,k)$, the flow conservation ensures that the inflow into $(k+1,k)$ is just enough to pass $1-\Pi_k$ onward while depositing just enough flow into $[k]$ to satisfy the required π_k. Therefore, any available solution to the original problem must be translatable to a solution for the restricted problem by a simple redistribution of flows. □

Essentially, Theorem 4 states that for any feasible solution that serves a user k impressions over $t \geq k$ visits, there must exist an alternative solution in which all k impressions are served upon the first k visits. Note that this structure in the solution produced by the restricted model implies that the resulting serving mechanism does not spread impressions served to a user uniformly throughout the serving period; a property which is typically desired by advertisers. Hojjat et al. (2016) develop a probabilistic model (see Appendix D of Chapter 1) that serves R&F contracts using patterns and show that in the absence of a pacing metric, a probabilistic model will arrange all impressions of the same ad in sequential slots of the pattern with no spreading. This conservative solution structure, which will be the most robust to randomness in user arrivals, is exactly what is produced here by our restricted version of the problem. Later in this section, we will discuss that the restricted model can also be much more easily extended to plan and serve multiple ad campaigns since the distribution of *leftover* user impressions are easy to characterize.

Theorem 4 is very important, because it establishes that a necessary and sufficient feasibility condition for the restricted problem is also a necessary and sufficient feasibility condition for the original problem. The restricted problem can be represented in a very compact form illustrated in Figure 2.3. In this representation, upon the k'th visit, an *active* user is in state $(k,k-1)$ and has been exposed to the ad $k-1$ times. With probability $1-x_{k,k-1}$, the publisher will not serve the user and marks him/her as *inactive* for all subsequent arrivals, causing the user to eventually be absorbed in state $[k-1]$. On the other hand, with probability $x_{k,k-1}$,

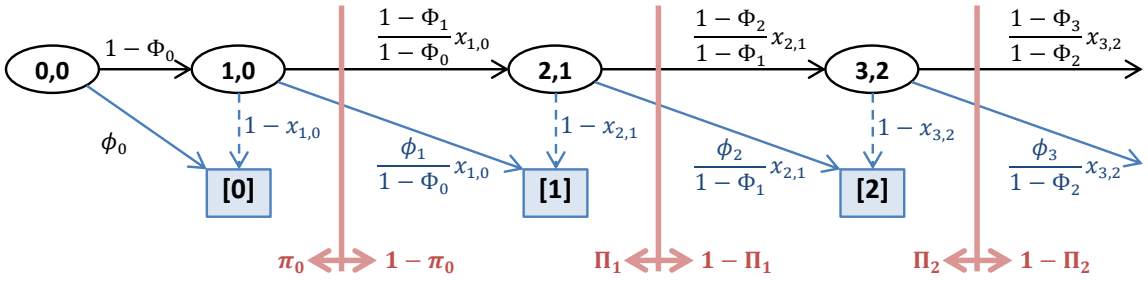


Figure 2.3: Simplified Markov Chain for the Fixed-Horizon Model

the publisher does serve the ad and the user remains active. Given that the user has made k visits, there is a $\phi_k/(1-\Phi_{k-1})$ chance that the user will never return and is absorbed in state $[k]$. Otherwise, with $(1-\Phi_k)/(1-\Phi_{k-1})$ probability, the active user will return in state $(k+1,k)$. This simplified Markov chain can be solved in closed form as follows.

- In state $(1,0)$, we need to have:

$$\phi_0 + (1-\phi_0)(1-x_{1,0}) \;=\; \pi_0 \qquad \rightarrow \qquad x_{1,0} \;=\; \frac{1-\pi_0}{1-\phi_0} \tag{2.13}$$

- In each state $(k+1,k)$, $k \geq 1$, assuming all preceding exposure rates are set correctly, there must be a total flow of $1-\Pi_{k-1}$ from $(k,k-1)$ which is split between $(k+1,k)$ and $[k]$ according to ratios $(1-\Phi_k)/(1-\Phi_{k-1})$ and $\phi_k/(1-\Phi_{k-1})$, respectively. Similarly, we want the total outflow from $(k+1,k)$ toward $(k+2,k+1)$ and $[k+1]$ to be $1-\Pi_k$. This outflow is simply the $x_{k+1,k}$ proportion of inflow into $(k+1,k)$. Therefore:

$$\Big(\frac{1-\Phi_k}{1-\Phi_{k-1}}\big(1-\Pi_{k-1}\big)\Big)x_{k+1,k} \;=\; 1-\Pi_k \quad \rightarrow \quad x_{k+1,k} \;=\; \frac{1-\Pi_k}{1-\Pi_{k-1}}.\frac{1-\Phi_{k-1}}{1-\Phi_k};\; \forall k \geq 1 \tag{2.14}$$

- It is trivial that $x_{k+1,k} = 0$ for all $k \geq F$, since the advertiser never wishes to have more than F impressions served to a user.

We can now characterize the necessary and sufficient condition for feasibility:

Theorem 5 (Necessary and Sufficient Condition for Feasibility)**.** *For the publisher to be able to implement the frequency distribution* $\boldsymbol{\pi} = (\pi_0, \pi_1, ..., \pi_F)$*, it is sufficient and necessary to*

have: $\pi_0 \geq \phi_0$, *and*

$$\forall k \in \{1, ..., F\}: \quad \frac{\pi_k}{1 - \Pi_{k-1}} \geq \frac{\phi_k}{1 - \Phi_{k-1}} \tag{2.15}$$

That is, the advertiser's requested frequency must be smaller than the number of user visits in the hazard rate order.

Proof. The feasibility condition is derived by simply imposing the probability rule $x_{k+1,k} \leq 1$ to equations (2.13) and (2.14). □

Hazard rate order implies first-order stochastic dominance. Note that $\pi_0 \geq \phi_0$ is established by the theorem. For every $k \geq 1$ we can show by induction that $\Pi_{k-1} \geq \Phi_{k-1}$ together with (2.15) implies $\Pi_k \geq \Phi_k$:

$$\begin{aligned}
\text{If } \Pi_{k-1} \geq \Phi_{k-1}: \quad &\rightarrow \quad \phi_k + \Pi_{k-1}(1 - \Phi_{k-1} - \phi_k) \geq \phi_k + \Phi_{k-1}(1 - \Phi_{k-1} - \phi_k) \\
&\rightarrow \quad \phi_k(1 - \Pi_{k-1}) \geq (1 - \Phi_{k-1})(\Phi_{k-1} + \phi_k - \Pi_{k-1}) \\
&\rightarrow \quad \frac{1 - \Pi_{k-1}}{1 - \Phi_{k-1}}\phi_k \geq \Phi_{k-1} + \phi_k - \Pi_{k-1} \\
(2.15): \quad &\rightarrow \quad \pi_k \geq \frac{1 - \Pi_{k-1}}{1 - \Phi_{k-1}}\phi_k \geq \Phi_{k-1} + \phi_k - \Pi_{k-1} \\
&\rightarrow \quad \Pi_{k-1} + \pi_k \geq \Phi_{k-1} + \phi_k \\
&\rightarrow \quad \Pi_k \geq \Phi_k.
\end{aligned}$$

Therefore, the necessary and sufficiency condition in Theorem 5 naturally implies the necessary condition previously established by Theorem 3. Note that each π_k requires users who visit k or more times. Theorem 5 states that upon each k'th visit, the advertiser's frequency distribution should be demanding a higher conditional no-return probability (resp., lower conditional return probability) compared to what is actually dictated by the user visit process. This is a stronger condition than first-order stochastic dominance.

2.3.3 Illustrative Examples

Assume that the advertiser wishes to reach 50% of the publisher's user traffic, of which $\pi'_1 = 0.3$ should be exposed once, $\pi'_2 = 0.6$ should be exposed twice, and $\pi'_3 = 0.1$ should be exposed three times to the ad by the end of the horizon. This translates into a frequency distribution $\boldsymbol{\pi} = (1-0.5, 0.3(0.5), 0.6(0.5), 0.(0.5)) = (0.5, 0.15, 0.3, 0.05)$, $\boldsymbol{\Pi} = (0.5, 0.65, 0.95, 1)$. Further assume that the publisher expects each user to make any number from zero to four visits throughout the campaign's horizon with uniform probabilities: $\boldsymbol{\phi} = (0.2, 0.2, 0.2, 0.2, 0.2)$, $\boldsymbol{\Phi} = (0.2, 0.4, 0.6, 0.8, 1)$.

Once can easily verify that the necessary and sufficient conditions stated in Theorem 5 are satisfied. Using equations (2.13) and (2.14) we obtain: $x_{1,0} = 5/8$, $x_{2,1} = 14/15$, $x_{3,2} = 3/14$, $x_{4,3} = 0$. That is, upon the first arrival we mark 3/8 of users as inactive and never serve them with an ad. We know that 80% of users actually make any first visit, 3/8 of which corresponds to 30% of the overall traffic. Together with the 20% who never make any visit, we keep a total of $\pi_0 = 50\%$ of users in state [0]. Among those who make a first visit, $\phi_1/(1-\Phi_0) = 0.25$ never make a second visit. Therefore, by serving ad to 5/8, i.e., 50% of the traffic, $(0.25)(50\%) = 12.5\%$ directly get absorbed in state [1], and the remaining 37.5% make a second visit as active users. Of these, we mark 1/15 (i.e., 2.5%) as inactive which will complete the $\pi_2 = 15\%$ of users who should be kept in state [1]. Of the remaining $14/15 \times 37.5\% = 35\%$ portion of user traffic to which we show the ad and keep as active, $\phi_2/(1-\Phi_1) = 1/3$ never make a third visit. Therefore, 11.67% get absorbed in state [2]. Among the remaining 23.33% who make a third visit, $(11/14)(23.33) = 18.33\%$ will be marked as inactive and kept in state [2] which will complete a total of $\pi_2 = 30\%$. Finally, the remaining $(3/14)(23.33) = 5\%$ who will be served an ad will all end up in absorbent state [4], either directly by making no additional visits, or by visiting a fourth time but not being served any further impressions. The flow diagram for this example is presented in Figure 2.4.

When Theorem 5 holds, a feasible solution to the problem exists and one such solution

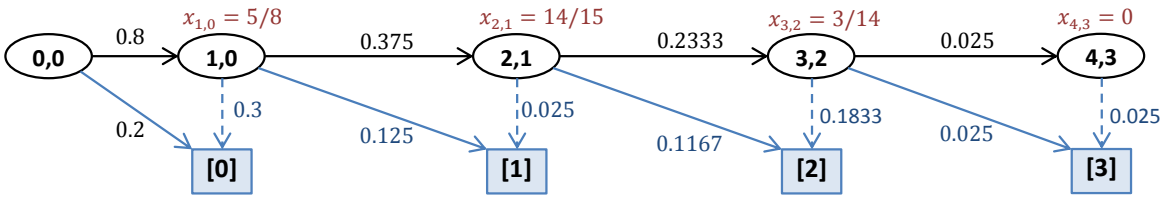


Figure 2.4: Illustrative Example for the Simplified Fixed-Horizon Markov Chain $\boldsymbol{\pi} = (0.5, 0.15, 0.3, 0.05)$, $\boldsymbol{\phi} = (0.2, 0.2, 0.2, 0.2, 0.2)$.

Table 2.1: Alternative Solution to the Example using the Complete Markov Chain

$x_{t,k}$	$k = 0$	1	2	3
$t = 1$	.4674	–	–	–
2	.3822	.8520	–	–
3	.0301	1	.2198	–
4	0	1	.1284	0

comes from (2.13) and (2.14). However, this solution is often unique. Using the the math program defined in §2.3.2.1, we could search for other solutions. For instance, using an objective which minimizes the sum of squared flows, e.g., to encourage some spread among impressions served to each user across time, we instead obtain the solution shown in Table 2.1.

Now consider the case in which the advertiser wishes to have $\boldsymbol{\pi} = (0.5, 0.05, 0.3, 0.15)$, in which π_1 and π_3 values are swapped. Even though the cumulative distribution $\boldsymbol{\Pi} = (0.5, 0.55, 0.85, 1)$ satisfies the first-order stochastic dominance condition from Theorem 3, it does not satisfy the sufficient condition of Theorem 5. In particular, $\pi_2/(1 - \Pi_1) = 0.1$ is less than $\phi_2/(1 - \Phi_1) = 0.25$. Therefore, the problem is infeasible. One could verify, e.g., by solving the math program from §2.3.2.1, that the original problem is also infeasible in this case.

2.3.4 Model Extensions

We now investigate two useful extensions of the model. First, we show that under the restricted model, it is easy to serve additional ad campaigns in a priority order. Then we

discuss the case in which publisher's traffic is composed to different user classes, each of which follow a different visit probability distribution.

2.3.4.1 Multiple Ad Campaigns

Simultaneous planning of multiple campaigns using the Markov Chain approach suffers from the curse of dimensionality. However, in practice, one can imagine that requests for advertising campaigns arrive sequentially, and the publisher evaluates the feasibility of serving each new campaign given the set of existing campaigns. The solution structure that is derived from the simplified Markov Chain makes it very easy to characterize the visit distribution of excess traffic once an existing campaign is served. Let $\tilde{\boldsymbol{\phi}} = (\tilde{\phi}_0, ..., \tilde{\phi}_L)$ denote this distribution. It is convenient to define $\tilde{\boldsymbol{\phi}}_+ = (\tilde{\phi}_1, ..., \tilde{\phi}_L)$ to denote the portion of this distribution corresponding to non-zero arrivals.

Assume the publisher is serving a first-priority campaign with a desired frequency distribution $\boldsymbol{\pi} = (\pi_0, ..., \pi_F)$ which satisfies the sufficient feasibility conditions of Theorem 5. It is easy to see in Figure 2.3 that:

- Upon the first visit, a $\pi_0 - \phi_0$ proportion of users are marked inactive and never served to the incumbent campaign. Therefore, their first and all subsequent arrivals can be allocated to a second campaign. These arrivals will follow the conditional distribution:

$$\tilde{\boldsymbol{\phi}}_{1+} = \left(\frac{\phi_1}{1 - \Phi_0}, \ ..., \ \frac{\phi_L}{1 - \Phi_0} \right)$$

- For each $k \in \{2, ..., F+1\}$, a $\pi_{k-1} - \frac{\phi_{k-1}}{1 - \Phi_{k-2}}(1 - \Pi_{k-2})$ proportion of users are marked as inactive upon their k'th visit, and therefore their k'th and subsequent arrivals can be allocated to a second campaign. These arrivals will follow the conditional distribution:

$$\tilde{\boldsymbol{\phi}}_{k+} = \left(\frac{\phi_k}{1 - \Phi_{k-1}}, \ ..., \ \frac{\phi_L}{1 - \Phi_{k-1}}, \ \overbrace{0, \ ..., \ 0}^{k-1} \right) \tag{2.16}$$

- The compound distribution of leftover traffic is therefore:

$$\tilde{\boldsymbol{\phi}}_+ \;=\; (\pi_0 - \phi_0)\tilde{\boldsymbol{\phi}}_{1+} \;+\; \sum_{k=2}^{F+1} \Big(\pi_{k-1} - \frac{\phi_{k-1}}{1-\Phi_{k-2}}(1-\Pi_{k-2})\Big)\tilde{\boldsymbol{\phi}}_{k+} \tag{2.17}$$

- To complete the distribution $\tilde{\boldsymbol{\phi}}$, it is enough to set $\tilde{\phi}_0$ such that the summation of probabilities in $\tilde{\boldsymbol{\phi}}$ adds up to 1.

Once $\tilde{\boldsymbol{\phi}}$ is found, the publisher may investigate whether the (reach adjusted) frequency distribution of a second campaign satisfies the sufficiency condition of Theorem 5 with respect to the leftover traffic distribution $\tilde{\boldsymbol{\phi}}$. If so, the exposure rates for the second campaign would be calculated using (2.13) and (2.14). Once a user is marked inactive for the incumbent campaign, he/she will immediately be marked active in state $(1,0)$ for the second campaign. The exact same calculation in (2.17) can be used to derive the leftover visit distribution for a possibility of serving a third campaign, and so on.

Example: In the example of Figure 2.4, the visit distribution of leftover traffic is given by:

$$\begin{aligned}
\tilde{\boldsymbol{\phi}}_+ \;&=\; 0.3 \times \Big(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\Big) + 0.025 \times \Big(\frac{1}{3},\frac{1}{3},\frac{1}{3},0\Big) + 0.1833 \times \Big(\frac{1}{2},\frac{1}{2},0,0\Big) + 0.025 \times \Big(1,0,0,0\Big) \\
&=\; \big(0.2,\; 0.1750,\; 0.0833,\; 0.0750\big) \\
\Rightarrow \qquad \tilde{\boldsymbol{\phi}} \;&=\; \big(0.4667,\; 0.2,\; 0.1750,\; 0.0833,\; 0.0750\big)
\end{aligned}$$

In this case, a second campaign with frequency distribution $\boldsymbol{\pi}' = (0.5, 0.25, 0.25)$ satisfies the feasibility conditions of Theorem 5 and can be implemented. Such campaign is also aiming to reach 50% of publisher's user traffic, with equal proportions exposed to 1 and 2 ad impressions. The exposure rates for the second campaign using the simplified Markov Chain structure will be: $x'_{1,0} = 0.9375$, $x'_{2,1} = 0.8$, $x'_{3,2} = 0$, per equations (2.13) and (2.14). The leftover visit distribution will be $\tilde{\boldsymbol{\phi}}' = (0.7854,\; 0.1078,\; 0.0828,\; 0.0193,\; 0.0047)$ per equation (2.17). We can then implement a third campaign with $\boldsymbol{\pi}'' = (0.8, 0.15, 0.05)$, which reaches 20% of user

traffic with 2/3 seeing the ad once, and 1/3 seeing it twice. We will have $x''_{1,0} = 0.932$, $x''_{2,1} = 0.5024$, $x''_{3,2} = 0$. Note that 95% of publisher's impression traffic is captured by the above three campaigns.

Interestingly, the order in which ad campaigns are prioritized affects the feasibility of implementation. In the example above, if we reverse the order of implementation for the above three campaigns, we will find that the leftover visit distribution following the implementation of $\boldsymbol{\pi}''$ and $\boldsymbol{\pi}'$ is not good enough to allow for an implementation of $\boldsymbol{\pi}$. Therefore, if multiple campaign requests arrive simultaneously, the publisher may want to investigate all possible orderings of their implementation to (hopefully) arrive at a feasible solution.

Finally, we should point out that the leftover visit distribution $\tilde{\boldsymbol{\phi}}$ characterized by (2.17) is specific to the restricted Markov model. It is quite hard to derive such distribution for the general model presented in §2.3.1. Indeed, when $\boldsymbol{x}$ is specified by a math program solution rather than (2.13) and (2.14), $\tilde{\boldsymbol{\phi}}$ no longer solely depends on$\boldsymbol{\phi}$ and $\boldsymbol{\pi}$ and is tied to the exposure rates used throughout the chain to implement $\boldsymbol{\pi}$. For example, one could verify by simulation that the alternative solution to the example provided earlier in Table 2.1 leads to a leftover traffic that follows $\tilde{\boldsymbol{\phi}} = (0.3905,\ 0.3109,\ 0.1712,\ 0.0638,\ 0.0636)$ which is different from what we derived above for the restricted Markov chain. In fact, the leftover traffic from this alternative solution is unable to accommodate the second campaign $\boldsymbol{\pi}'$ in our example. By creating an allocation plan in which each frequency k is served using the first k visits from each user (i.e., the least number of visits possible), the solution to the restricted chain is not only the most robust to misspecification of the arrival distribution, but also leaves the most predictable and stable leftover traffic to be allocated to lower-priority campaigns.

In the general Markov chain model, a user may always remain *active* with respect to the incumbent campaign. A decision of not showing the ad upon a visit provides an opportunity for serving other campaigns but gives little clue as to how subsequent visits from the user may become available to a secondary campaign as there remain numerous possibilities for the

future sample path of that user within the chain. We leave further exploration of this case to future research.

2.3.4.2 Multiple User Types

We now address the case in which the publisher's traffic is non-homogeneous. In particular, we assume that users can be clustered in V groups. The number of visits from users in group $v = \{1, ..., V\}$ are i.i.d. and follow the distribution $\boldsymbol{\phi}^{(v)} = \left(\phi_0^{(v)}, \phi_1^{(v)}, ..., \phi_L^{(v)}\right)$. We assume the publisher can identify these distributions and as well identify the cluster to which each user belongs. Furthermore, it is known that an α_v proportion of users belong to cluster v. Indeed, we must have $\sum_{v=1}^{V} \alpha_v = 1$, and $\sum_{t=0}^{L} \phi_t^{(v)} = 1$ for each $v \in \{1, ..., V\}$.

Clearly, the publisher has the option of ignoring this information and modeling user visits as i.i.d. from the aggregate/compound distribution $\bar{\boldsymbol{\phi}} = \sum_v \alpha_v \boldsymbol{\phi}^{(v)}$. If this distribution satisfies the sufficiency conditions of Theorem 5 with respect to an advertiser's requested frequency distribution $\boldsymbol{\pi}$, the publisher can proceed by treating all users equally and obtain the appropriate exposure rates using (2.13) and (2.14).

An interesting problem arises when the compound distribution $\bar{\boldsymbol{\phi}}$ does not meet the sufficiency conditions of Theorem 5 with respect to $\boldsymbol{\pi}$. The question is whether the publisher could benefit from treating users of each class v differently, i.e., by solving a separate restricted Markov chain model for each user class, so that a frequency distribution $\boldsymbol{\pi}$ is attained in aggregate. To this end, the publisher needs to break down the frequency distribution $\boldsymbol{\pi}$ into multiple frequency distributions, $\boldsymbol{\pi}^{(v)}$, one for each user class, such that:

- Each $\boldsymbol{\pi}^{(v)}$ is a probability distribution:

$$\sum_{k=0}^{F} \pi_k^{(v)} = 1, \qquad \forall v \tag{2.18}$$

$$\pi_k^{(v)} \geq 0, \qquad \forall k, v \tag{2.19}$$

- The frequency distribution $\boldsymbol{\pi}^{(v)}$ assigned to each user class v meets the sufficiency conditions of Theorem 5:

$$\pi_0^{(v)} \geq \phi_0^{(v)} \qquad \forall v \tag{2.20}$$

$$\pi_k^{(v)} \geq \frac{\phi_k^{(v)}}{1-\Phi_k^{(v)}}\Big(\sum_{k'=k}^{F} \pi_{k'}^{(v)}\Big) \qquad \forall v,\ k \in \{1,...,F\} \tag{2.21}$$

- The aggregate/compound frequency distribution should match the advertiser's specification:

$$\sum_v \alpha_v \pi_k^{(v)} = \pi_k \qquad \forall k \tag{2.22}$$

If a set of $\pi_k^{(v)}$ values exist that satisfy the set of linear inequalities (2.18)–(2.22), the publisher can benefit from his knowledge of user classes and implement $\boldsymbol{\pi}$ by treating each user class differently. Since we typically do not expect too many user classes (V) or very high frequency requirements (F), solving the above system of $O(VF)$ equations can be done very quickly using any standard math programming software.

Example: Continuing the example of §2.3.3, assume that the advertiser wishes to have a frequency distribution $\boldsymbol{\pi} = (0.5, 0.15, 0.3, 0.05)$. This time suppose that the user traffic is clustered in two groups: $\boldsymbol{\phi}^{(1)} = (0.4, 0.3, 0.2, 0.1, 0)$ who are relatively low-visiting, and $\boldsymbol{\phi}^{(2)} = (0, 0.1, 0.2, 0.3, 0.4)$ who can be considered as high visiting. If the traffic was split evenly between the two clusters, i.e., $\alpha_1 = \alpha_2 = 0.5$, the aggregate/compound visit distribution $\bar{\boldsymbol{\phi}} = (0.2, 0.2, 0.2, 0.2, 0.2)$, as shown previously, would satisfy the sufficiency conditions of Theorem 5. Hence, the publisher could apply the same solution from §2.3.3 to all users homogeneously, regardless of their type, in order to implement $\boldsymbol{\pi}$.

However, suppose $\alpha_1 = 0.7$ and $\alpha_2 = 0.3$. That is, a higher proportion of users are low-visiting. In this case, the compound distribution $\bar{\boldsymbol{\phi}} = (0.28, 0.24, 0.20, 0.16, 0.12)$ does not meet the sufficiency conditions of Theorem 5. In particular, $\pi_1/(1-\Pi_0) = 0.3$ is less than

$\bar{\phi}_1/(1-\bar{\Phi}_0) = 1/3$. Therefore, the publisher cannot implement $\boldsymbol{\pi}$ by treating all users equally. However, there exists a disaggregation of $\boldsymbol{\pi}$ among user classes which makes this possible. Using an LP-solver, we obtain $\boldsymbol{\pi}^{(1)} = (0.71, 0.15, 0.12, 0.02)$ and $\boldsymbol{\pi}^{(2)} = (0.01, 0.15, 0.72, 0.12)$. It is easy to verify that each of the two pairs $\boldsymbol{\pi}^{(v)}$ and $\boldsymbol{\phi}^{(v)}$ satisfies the sufficiency conditions and $\alpha_1\boldsymbol{\pi}^{(1)} + \alpha_2\boldsymbol{\pi}^{(2)} = \boldsymbol{\pi}$. The exposure rates for the restricted Markov model are given by $\boldsymbol{x}^{(1)} = (0.4833, 0.9655, 0.4286, 0)$ and $\boldsymbol{x}^{(2)} = (0.99, 0.9428, 0.1837, 0)$, elements respectively showing the ad display probability upon the first four arrivals.

It is easy to show that with higher ratio of low-visiting users, e.g., $\alpha_1 = 0.8$ and $\alpha_2 = 0.2$, the set of linear inequalities (2.18)–(2.22) has no feasible solution and user discrimination cannot benefit the publisher in implementing $\boldsymbol{\pi}$.

2.4 Rolling-Horizon Frequency Specification

In our fixed-horizon model, we showed that the publisher may employ a simple Markov chain model to serve advertising such that the proportion of viewers who are exposed to the ad a certain number of times throughout the campaign's horizon follows a particular distribution specified by the advertiser. The mechanism implemented by the simplified fixed-horizon model has the property that it serves all impressions of the same ad to each viewer upon successive visits. Hence, it does not provide a smooth delivery of campaigns over time, unless the arrival process to publisher's website is such that *initial* visits from different users are naturally spread uniformly throughout the campaign's horizon – a condition which may not hold in practice.

In this section, we introduce and study a different format in which an advertiser may wish to specify exposure frequency. Instead of counting the number of ad exposures to each customer over the entire campaign's horizon or non-overlapping fixed-length time intervals that span throughout the campaign's horizon (e.g., each calendar week), we could measure the number of ad exposures to each customer on a rolling basis, i.e., throughout a timespan

T (e.g., 24 hours) immediately preceding the current time, which we henceforth refer to as the *frame*. In other words, instead of resetting exposure counts to zero at the start of every T interval, we gradually erase the record of impressions that occurred outside the frame, i.e., more than T time units in the past.

In this new form of reach and frequency contract, the frequency distribution $\boldsymbol{\pi} = (\pi_0, ..., \pi_F)$ specified by the advertiser can be interpreted in two ways: 1) A π_k-proportion of users should be exposed to the ad k times in any randomly-selected time interval of length T; and 2) For a π_k-fraction of the time, we should observe each user (who is to be reached) to have seen the ad k times in the past T units of time.

In our fixed-horizon model, the frequencies were being counted over a static time interval and hence we could work with a probability distribution $\boldsymbol{\phi} = (\phi_0, \phi_1, ..., \phi_L)$ that described the number of page visits from users within that time. For our rolling horizon model, we need a more detailed characterization of the arrival process which further describes how/when user visits occur throughout the campaign's horizon. We assume that user visits are independent and identically distributed, following a Poisson distribution with constant exogenous rate λ per unit time.

Again, upon each user visit the publisher must decide whether or not to show an ad so that the reach and frequency requirements are met. In the remainder of this section we provide a solution to publisher's problem using a birth-and-death Markov chain model and characterize the feasibility criteria.

2.4.1 Discrete Time Markov Model

The frequency of exposures to each individual at any time t_0 is measured as the number of exposures delivered within a timespan of length T preceding t_0, i.e. within $[t_0 - T, t_0]$, which we refer to as the *frame*.

We partition the frame into n equal time periods of length $\delta t = T/n$. Denoting the

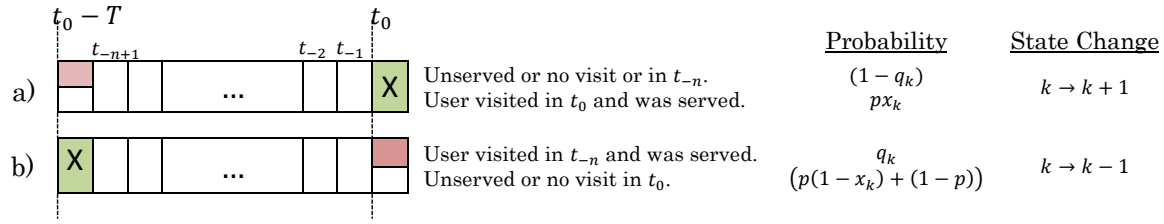


Figure 2.5: Frame, Time Periods, and State Transitions in the Rolling Horizon Markov Model

start of the current time period as t_0, we use t_{-i} to index the time period that starts at time $t_0 - (i)(\delta t)$. Therefore, t_{-1} is the most recent period $[t_0 - \delta t, t_0)$ and t_{-n} is the oldest period in the frame $[t_0 - T, t_0 - T + \delta t)$ for which we keep track of exposure frequencies. See Figure 2.5.

We assume n is large enough (resp., δt is small enough) that the probability of two or more visits in each period is negligible. Such discritization is a common approach in modeling dynamic allocation problems such as dynamic revenue optimization (e.g., see Lautenbacher and Stidham Jr 1999, as well as Talluri and Van Ryzin 2006, p.58). Essentially, the Poisson arrival process is approximated by a binomial process with a visit probability $p = \lambda \delta t$, which is constant across time and independent from one period to another. Clearly, $np = \lambda T$ which is constant. When $n \to \infty$ (resp., $\delta t \to 0$) this binomial process converges to the original Poisson process with rate λ.

The state of an individual at the beginning of t_0 can be indexed by $k \in \{0, 1, ..., n\}$ which shows the number of times the user has seen the ad over the past n time periods, i.e., in the current frame $[t_0 - T, t_0]$. Let $I(t_{-i}) = 1$ if an ad has been served in time period t_{-i} and zero otherwise. Then the state of a user at t_0 can be written as $S(t_0) = \sum_{i=1}^{n} I(t_{-i})$.

Upon any user visit, the publisher will display the ad to the user with probability $x_k \in [0, 1]$ if the individual is in state k. WLOG, we assume n is large enough that the advertiser never wants a frequency more than n. More precisely, we assume that the adviser has a frequency cap $F < n$, and therefore $x_k = 0$ for all $k \geq F$.

Each δt, the frame moves one time period forward. If an impression is served in t_0 it

will be added to the state $S(t_0)$ and if an impression has been served in t_{-n} it will exit the frame and will be subtracted from $S(t_0)$. That is, the state will be updated as $S(t_0) \leftarrow S(t_0) + I(t_0) - I(t_{-n})$. In order to perfectly track the state of the user note that the publisher would need to keep a record of the exact timestamps at which each user has arrived and been served an ad. For a large publisher with millions of user visits per hour, we expect the storing of information at such detail to be prohibitive. Therefore, we propose an alternative counting mechanism which proves to work quite effectively in our numerical experiments. Assume that the publisher does not maintain an exact record of times that ads have been served. Given that the current state is k, the publisher can use an estimate of the probability that an impression has been served at the beginning of the frame, denoted $q_k = P(I(t_{-n}) = 1 | S(t_0) = k)$, to subtract the frequency count probabilistically. Obviously, we should have $q_0 = 0$ (when the frame is empty) and $q_n = 1$ (when every time period in the frame contains an ad). Suppose there are k impressions in the frame. Regardless of how those k impressions are spread over the n slots of the frame, it is clear that if the publisher serves no further ads to the user, the frame should clear and the user's state should return to zero after n time periods (i.e., when T time is passed). By clearing the state at a rate $q_k = k/n$ we ensure that this happens. This is equivalent to defining a lifetime of T with uniform decay rate of $1/n$ for each impression served.

Note that the publisher need not update the state after every δt. Updating the state occurs only at points in time where the user makes an arrival. The publisher only needs to know (i.e., store) the timestamp of the previous arrival along with the updated state at that time. The publisher can then simulate the decay process since the last user visit to arrive at an estimate for the current state. If the user was last seen in t_{-i}, i.e., $(i)(\delta t)$ time ago and left in (post-exposure) state $\bar{S}(t_{-i}) + I(t_{-i}) = k$, the new estimated state state will be given by $\bar{S}(t_0) = (k - D)^+$ in which $D \sim \text{Binom}(i, q_k)$ is a binomial random variable with i trials and success probability q_k. Following the publisher's exposure decision, the user will be left

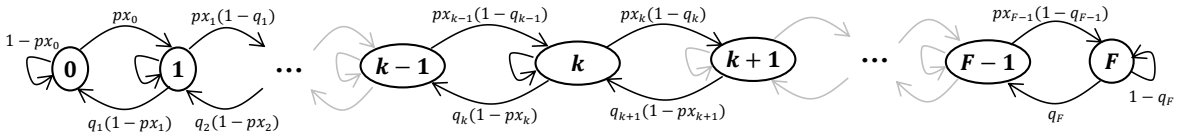


Figure 2.6: States and Transition Probabilities in the Rolling-Horizon Markov Chain

in state $\bar{S}(t_0) + I(t_0)$ until next arrival.

We now characterize the state transition probabilities. Three events should occur for a user to switch state from k to $k+1$: The user should make a visit (p probability), the publisher should choose to display the ad upon that visit (x_k probability), and it must be that there was no ad shown to the user at t_{-n} ($1-q_k$ probability); See Figure 2.5(a). Similarly, for a user to transition from state k to $k-1$ it must be that the publisher showed an ad at t_{-n} (q_k probability) and no ad is shown at the current time period, either because of no visit from the user ($1-p$ probability) or a visit upon which the publisher chose not to serve an ad ($p(1-x_k)$ probability); See Figure 2.5(b). Therefore, if P denotes the transition matrix: $P_{k,k+1} = px_k(1-q_k)$ and $P_{k,k-1} = q_k(1-p+p(1-x_k)) = q_k(1-px_k)$. The user will remain at state k with probability $P_{k,k} = 1 - P_{k,k+1} - P_{k,k-1}$.

User states and transition probabilities are shown in Figure 2.6. It is clear and rather expected that the Markov chain of the rolling-horizon reach and frequency problem models a birth and death process. Detailed balance equations are as follows:

$$q_k(1-px_k)\pi_k \;=\; px_{k-1}(1-q_{k-1})\pi_{k-1} \qquad \forall k = 1,\ldots,F \tag{2.23}$$

Recall that $q_0 = 0$, $q_n = 1$, $F < n$ and $x_k = 0$ for all $k \geq F$. To further justify the choice of $q_k = k/n$, note that we can rearrange the detailed balance equations as:

$$p(1-q_{k-1})\; x_{k-1}\pi_{k-1} \;+\; pq_k\; x_k\pi_k \;=\; q_k\pi_k.$$

Taking the summation of both sides over $k \in \{1, ..., F\}$, we obtain:

$$\sum_{k=0}^{F} px_k \pi_k \;=\; \sum_{k=0}^{F} q_k \pi_k, \tag{2.24}$$

which indicates that the average serving rate should equal the average exit rate. This is intuitive, because impressions that enter and exit the frame follow the exact same pattern, with a lag of n time periods. Now note that per the advertiser's request, the user is expected to be in state k with probability π_k. Therefore, $\sum_{k=0}^{F} k\pi_k$ (the mean of the $\boldsymbol{\pi}$ distribution) gives the average number of impressions we should expect to observe in any randomly-chosen frame. On the other hand, when n is large, each randomly-selected frame by itself constitutes a large-enough sample so that we can expect the user to be in state k in $n\pi_k$ time periods of the frame. That is, we should expect the user to be served at the rate px_k in $n\pi_k$ time periods of any randomly-selected frame. Therefore, $\sum_{k=0}^{F}(n\pi_k)(px_k)$ also gives the average number of impressions expected in such frame. Hence, we must have:

$$n\sum_{k=0}^{F} px_k \pi_k \;=\; \sum_{k=0}^{F} k\pi_k. \tag{2.25}$$

From (2.24) and (2.25) we obtain:

$$\sum_{k=0}^{F} q_k \pi_k \;=\; \sum_{k=0}^{F} \frac{k}{n}\pi_k. \tag{2.26}$$

Even though (2.26) does not uniquely define each q_k, it suggests that $q_k = k/n$ is a valid candidate.

Given p, n, and ad serving probabilities x_k, we can find the stationary frequency distribution $\boldsymbol{\pi} = (\pi_0, ..., \pi_F)$ as follows: Pick an initial value for π_0 (strictly above zero). Calculate π_1 through π_F recursively, using:

$$\pi_k \;=\; \left(\frac{n+1}{k} - 1\right) \frac{px_{k-1}}{1 - px_k}\, \pi_{k-1} \qquad \forall k = 1, \ldots, F \tag{2.27}$$

which is obtained by replacing $q_k = k/n$ in (2.23) and simple rearrangement. To enforce

$\sum_{k=0}^{F} \pi_k = 1$, divide each value by the current total summation of π_k. This normalization step is correct because (2.27) can be expanded to write each π_k as $C_k\pi_0$; hence, $\sum_k \pi_k = (\sum_k C_k)\pi_0$. Therefore, any scaling applied to π_0, will directly impact all other π_k and the total sum $\sum_k \pi_k$ by the same factor.

The publisher's decision is to find the ad serving probabilities x_k given p, n, and a desired stationary frequency distribution $\boldsymbol{\pi} = (\pi_0, ..., \pi_n)$. To this end, set $x_F = 0$. Then calculate x_{F-1} through x_0 recursively (backwards), using:

$$x_k \;=\; \frac{k+1}{n-k}\frac{\pi_{k+1}}{\pi_k}\left(\frac{1}{p} - x_{k+1}\right) \qquad \forall k = 0, \ldots, F-1 \tag{2.28}$$

which is obtained by a simple rearrangement of (2.27). If any x_k turns out to be outside $[0, 1]$, we conclude that the stationary distribution π_k cannot be achieved with the given arrival rate and frame size. Since the feasibility condition for x_k depends on the value of x_{k+1}, assessing whether a desired distribution $\boldsymbol{\pi}$ can be implemented is equivalent to attempting to solve for serving probabilities using the recursive equations (2.28), which can be done in $O(F)$ time.

Theorem 6 (Sufficient Condition for Feasibility)**.** *For the publisher to be able to implement the rolling-horizon frequency distribution $\boldsymbol{\pi} = (\pi_0, \pi_1, ..., \pi_F)$, it is sufficient to have:*

$$(k+1)\frac{\pi_{k+1}}{\pi_k} \;\leq\; \lambda T\Big(1 - \frac{k}{n}\Big) \qquad \forall k = 0, \ldots, F-1. \tag{2.29}$$

Proof. The condition is obtained by substituting (2.28) into the feasibility condition $x_k \leq 1$:

$$\frac{k+1}{n-k}\frac{\pi_{k+1}}{\pi_k}\left(\frac{1}{p} - x_{k+1}\right) \;\leq\; \frac{k+1}{n-k}\frac{\pi_{k+1}}{\pi_k}\frac{1}{p} \;\leq\; 1 \qquad \forall k = 0, \ldots, F-1$$

Rearranging terms, and noting that $p = \lambda\delta t = \lambda T/n$ will give (2.29). Note that with $p \leq 1$, (2.29) implies:

$$(k+1)\frac{\pi_{k+1}}{\pi_k} \leq n-k \quad \rightarrow \quad \frac{k+1}{n-k}\frac{\pi_{k+1}}{\pi_k} \leq 1 \quad \rightarrow \quad x_k = \frac{k+1}{n-k}\frac{\pi_{k+1}}{\pi_k}\left(\frac{1}{p} - x_{k+1}\right) \leq \frac{1}{p} - x_{k+1} \leq \frac{1}{p},$$

which ensures non-negativity of x_{k-1}. □

2.4.2 Continuous Time Markov Model

If we consider infinitely small divisions of time, i.e., take the limit $\delta t \to 0$, (resp., $n \to \infty$), we know that the binomial arrival process defined in §2.4.1 converges to a Poisson process with constant rate λ per unit time. The quantity $np = \lambda T$ remains constant, while $p \to 0$. In this case, (2.28) converges to:

$$\lim_{\delta t \to 0} x_k \;=\; \tilde{x}_k \;=\; \frac{k+1}{\lambda T}\frac{\pi_{k+1}}{\pi_k} \qquad \forall k = 0, \ldots, F-1, \tag{2.30}$$

where λT is the expected number of arrivals from the user over the timespan of the frame.

To see the error of approximating x_k with $\bar{x}_k$, note that (2.28) can be written as:

$$x_k \;=\; \left(\frac{1 - p x_{k+1}}{1 - k/n}\right)\tilde{x}_k \;=\; \left(\frac{n - \lambda T x_{k+1}}{n-k}\right)\tilde{x}_k \qquad \forall k = 0, \ldots, F-1. \tag{2.31}$$

The ratio $\frac{1-px_{k+1}}{1-k/n}$ varies between $1-p$ (when $k=0$ and $x_{k+1}=1$) to $1+\frac{F}{n-F}$ (when $k=F$ and $x_{k+1}=0$). Therefore,

$$\left|x_k - \tilde{x}_k\right| \;\leq\; \max\left\{p,\; \frac{F}{n-F}\right\}\tilde{x}_k. \tag{2.32}$$

This also implies that the approximation must be accurate when $F \ll n$.

The feasibility condition $x_k \leq 1$ simplifies to:

$$(k+1)\frac{\pi_{k+1}}{\pi_k} \;\leq\; \lambda T \qquad \forall k = 0, \ldots, F-1, \tag{2.33}$$

which can be interpreted as a lower-bound requirement on the arrival rate λ.

Theorem 7 (Necessary Condition for Feasibility). *For the publisher to be able to implement*

the rolling-horizon frequency distribution $\boldsymbol{\pi} = (\pi_0, \pi_1, ..., \pi_F)$*, it must be that:*

$$\pi_0 \geq \frac{\phi_{\lambda T}(0)}{\Phi_{\lambda T}(\bar{f})} \qquad \& \qquad \frac{\pi_k}{\pi_0} \leq \frac{\phi_{\lambda T}(k)}{\phi_{\lambda T}(0)} \qquad \forall k = 1, \ldots, F, \tag{2.34}$$

where $\phi_{\lambda T}(k)$ *and* $\Phi_{\lambda T}(k)$ *are the Poisson PDF and CDF of observing* k *visits from the user over the timespan of the frame. Essentially, the probability ratio* π_k/π_0 *should not exceed the corresponding ratio in the (Poisson) distribution that describes the number of arrivals over the frame's timespan* T*.*

Proof. Since both sides of all inequalities in (2.33) are strictly positive, we can multiply the first k inequalities:

$$(2.33) \quad \Rightarrow \quad \frac{\pi_1}{\pi_0}\frac{2\pi_2}{\pi_1} \cdots \frac{k\pi_k}{\pi_{k-1}} = (k!)\frac{\pi_k}{\pi_0} \leq (\lambda T)^k \qquad \forall k = 1, \ldots, F$$

Moving $(k!)$ to the right hand side, and multiplying both sides by the constant $\phi_{\lambda T}(0) = e^{-\lambda T}$, we obtain:

$$\frac{e^{-\lambda T}}{\pi_0}\pi_k \leq \frac{e^{-\lambda T}(\lambda T)^k}{k!} = \phi_{\lambda T}(k) \quad \rightarrow \quad \frac{\pi_k}{\pi_0} \leq \frac{\phi_{\lambda T}(k)}{\phi_{\lambda T}(0)} \qquad \forall k = 1, \ldots, F$$

If the above holds, we must have:

$$\sum_{k=0}^{F} \frac{\pi_k}{\pi_0} \leq \sum_{k=0}^{F} \frac{\phi_{\lambda T}(k)}{\phi_{\lambda T}(0)} \quad \rightarrow \quad \frac{1}{\pi_0} \leq \frac{\Phi_{\lambda T}(F)}{\phi_{\lambda T}(0)} \quad \rightarrow \quad \pi_0 \geq \frac{\phi_{\lambda T}(0)}{\Phi_{\lambda T}(F)},$$

which gives a necessary condition for π_0 so (2.34) holds. □

2.4.3 Illustrative Examples

Assume that the frame is defined as a 24-hour period, and users visit publisher's website according to a Poisson process with mean $\lambda T = 20$. Suppose that the advertiser wishes to reach $\theta = 0.5$ of users with a rolling frequency distribution of $\boldsymbol{\pi} = (0.01, 0.09, 0.2, 0.5, 0.2)$. That is,

Table 2.2: Solution to the Rolling-horizon Example at Different Levels of Discritization
$\lambda T = 20$, $\boldsymbol{\pi} = (0.01, 0.09, 0.2, 0.5, 0.2)$

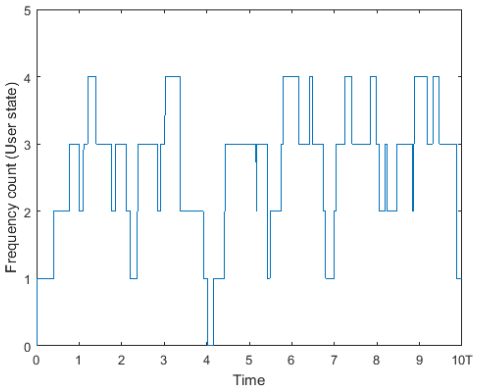
n	p	x_0	x_1	x_2	x_3	$x_{k\geq 4}$
200	.10	.4403	.2149	.3757	.0812	0
500	.04	.4461	.2193	.3753	.0805	0
1000	.02	.4480	.2208	.3751	.0802	0
5000	.004	.4496	.2219	.3750	.0800	0
∞	$-$	.4500	.2222	.3750	.0800	0

among randomly selected 24-hour periods throughout the campaign's horizon, $\{1, 9, 20, 50, 20\}\%$ of the time each user who is to be reached should be observed to have seen the ad $\{0, 1, 2, 3, 4, 5\}$ times, respectively. Using (2.28) and (2.30), we obtain the solutions presented in Table 2.2 for different levels of discritization, $n \in \{200, 500, 1000, 5000, \infty\}$. The example shows that the error of approximating a discrete-time solution with the continuous time solution can be quite small.

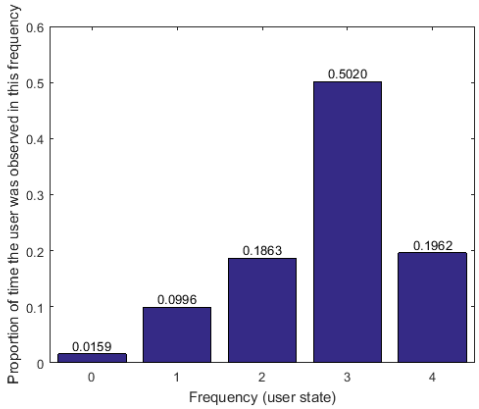
Figure 2.7 shows the performance of the solution obtained for $n = 1000$ which we simulated over a 10-day period (total duration of $10T$) for a single user. Figure 2.7(a) shows the progression of user's state (rolling-horizon frequency count) throughout the campaign's horizon. Figure 2.7(b) summarizes the empirical distribution of user frequency which is very close to the advertiser's requested $\boldsymbol{\pi}$ distribution.

Figure 2.7(c) shows the cumulative count of impressions served to the user over time. It is clear that the cumulative count is very close to the ideal target which grows uniformly over time at the average frequency rate $\sum_k k\pi_k$. This shows that the rolling-horizon model can provide an ad delivery mechanism which naturally enforces uniform delivery of a campaign over its horizon. Indeed, if n is large and the variance of the frequency distribution $\boldsymbol{\pi}$ is small, we should expect the rolling horizon model to deliver ads at a rate close to the mean $\sum_k k\pi_k$ as it reaches the steady state. However, if the variance of $\boldsymbol{\pi}$ is large, the ad delivery may not be smooth over time.

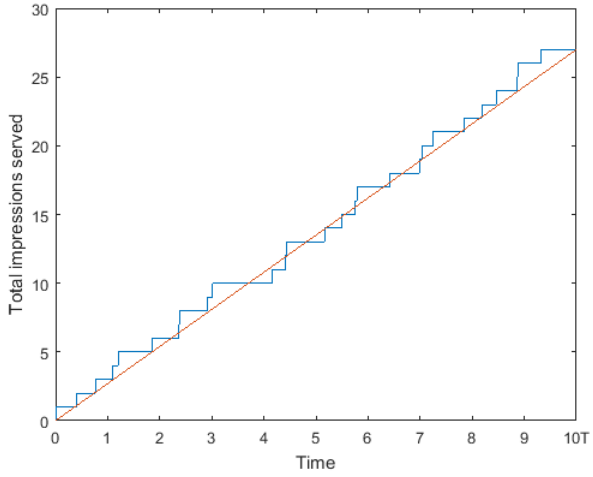
For an alternative example, assume $\lambda T = 20$ and $n = 1000$ and $\boldsymbol{\pi} = (0.4, 0.09, 0.02, 0.09, 0.4)$.



(a) User's state progression over time



(b) Empirical frequency distribution attained at the end of campaign's horizon



(c) Cumulative count of impressions served over time

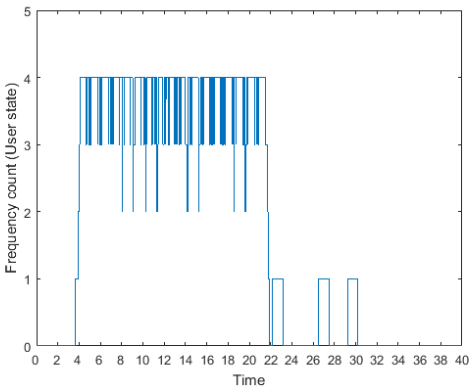
Figure 2.7: Simulation of the Rolling-horizon Markov Chain Model for a Single User $\lambda T = 20$, $n = 1000$, $\boldsymbol{\pi} = (.01, .09, .2, .5, .2)$, Campaign horizon $= 10T$.

Such frequency distribution traps users in either state 0 or $F = 4$ for most of the horizon. Therefore, it is natural to expect a non-smooth delivery over time. The solution $\boldsymbol{x} = (0.0112, 0.0219, 0.6643, 0.8915)$ is such that a user in state $k \geq 2$ is quickly pushed to $F = 4$ by high exposure rates upon any visit, whereas a user in state $k \leq 1$ is pushed toward state 0 by using very low exposure rates. Figure 2.8 shows the result of simulating this solution over a horizon of $40T$. As expected the delivery is not smooth (Figure 2.8a), but note that the above choice of $\boldsymbol{\pi}$ has effectively implemented the commonly used *pulsing* strategy in advertising (see Naik et al. 1998).

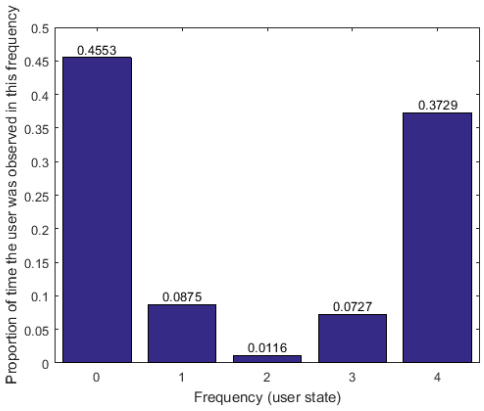
Note that the variance (or standard deviation) of $\boldsymbol{\pi}$ provides a measure of the degree to which the actual exposure rate, at any point in time, may differ from the ideal average $\sum_k k\pi_k$. It is hard to mathematically quantify the smoothness of delivery, e.g., as a sum of squared deviations between the sample path of the cumulative sum of impressions delivered over time and the ideal path with constant slope $\sum_k k\pi_k$. Measuring these deviations involves evaluating a nonlinear integral of the sample path of the Markov chain. The reader may refer to Puri (1966), McNeil (1970), and Pollett (2003) for examples of such derivations for basic integrals of birth and death processes and their properties. We leave the adaptation of such techniques for measuring non-smoothness of ad delivery to future research.

2.5 Conclusions

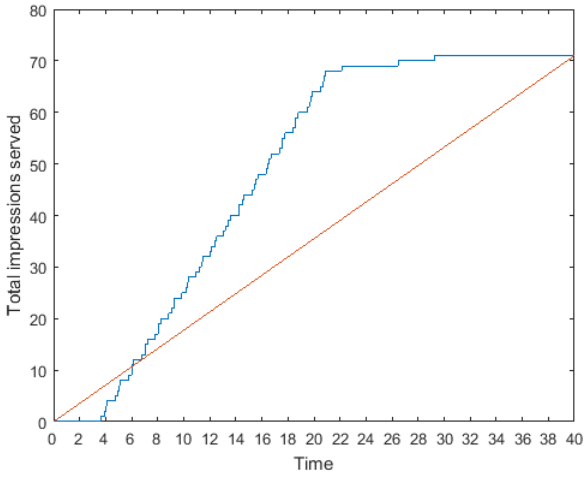
In this paper we introduced and studied two new variants of reach and frequency (R&F) contracts for online advertising in which the advertiser specifies the fraction of publisher's user traffic she wishes to reach (by at least 1 impression) and a frequency distribution which specifies what proportion of individuals should be exposed at what frequency. In our fixed-horizon variant, we assumed that frequencies are either counted throughout the entire campaign's horizon or the counts are reset after certain time units (e.g., at the start of every calendar week). In our rolling-horizon variant, we assumed that frequencies are measured on



(a) User's state progression over time



(b) Empirical frequency distribution attained at the end of campaign's horizon



(c) Cumulative count of impressions served over time

Figure 2.8: Simulation of the Rolling-horizon Markov Chain Model for a Single User Alternative Example: $\lambda T = 20$, $n = 1000$, $\boldsymbol{\pi} = (.4, .09, .02, .09, .4)$, Campaign horizon $= 40T$.

a rolling basis, e.g., over any randomly-selected 24-hour intervals throughout the campaign's horizon.

For each of the two variants, we developed a Markov chain model, characterized the feasibility criteria, and derived a state-dependent impression assignment rule for the publisher to satisfy R&F contract when the feasibility criteria are met. In particular, we developed a simplified Markov chain for the fixed-horizon contract for which the publisher's decision could be written in closed-form. This simplified model also allowed us to extend our approach to planning multiple ad campaigns or multiple user types with different arrival processes. Similarly, we developed a continuous-time Markov chain model for the rolling-horizon contract which allowed for obtaining a closed-form solution. We showed that in both models, obtaining the publisher's assignment rule can be done very efficiently in linear time in the length of the frequency distribution specified by the advertiser.

Finally, we discussed that the fixed horizon contract may lead to poor spreading of campaigns throughout their horizon whereas the rolling-horizon variant naturally attains uniform spearing of a campaign if the variance of the frequency distribution is relatively low.

We left several interesting directions to future research. For instance, in the fixed-horizon model, we showed that when multiple campaigns are implemented in a priority order, the feasibility of implementing R&F specifications of all campaigns may depend on the order in which campaigns are considered. However we did not characterize this order or possibly a more advanced Markov chain model which simultaneously plans multiple campaigns. Furthermore, we showed that the uniform spreading of campaigns in the rolling-horizon model is related to the variance of the frequency distribution specified by the publisher. However, we did not formalize this relationship. Finally, extensions of the rolling-horizon model to multiple campaigns and user visit types could be of interest to future research.

Reach and frequency contracts are becoming of increasing value to online advertisers and

we believe our modeling and solution approach can be quite useful in practice, and help toward a deeper understanding of the serving reach and frequency contracts.

Bibliography

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.

Archak, N., V. Mirrokni, and S. Muthukrishnan (2010). Budget optimization for online advertising campaigns with carryover effects. In *Sixth Ad Auctions Workshop*. Citeseer.

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.

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.

Briggs, R. and N. Hollis (1997). Advertising on the web: is there response before click-through? *Journal of Advertising research 37*, 33–46.

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.

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

Dreze, X. and F.-X. Hussherr (2003). Internet advertising: Is anybody watching? *Journal of interactive marketing 17*(4), 8–23.

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

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.

Hojjat, A., J. Turner, S. Cetintas, and J. Yang (2016). A unified framework for the scheduling of guaranteed targeted display advertising under reach and frequency requirements. Working paper.

Interactive Advertising Bureau (2016, April). IAB 2015 full-year internet advertising revenue report. Available online at: `http://www.iab.net/research/industry_data_and_landscape/adrevenuereport`.

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`.

Lautenbacher, C. J. and S. Stidham Jr (1999). The underlying markov decision process in the single-leg airline yield-management problem. *Transportation Science 33*(2), 136–146.

McNeil, D. R. (1970). Integral functionals of birth and death processes and related limiting distributions. *The Annals of Mathematical Statistics 41*(2), 480–485.

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

Morimura, T., T. Osogami, and T. Idé (2013). Solving inverse problem of markov chain with partial observations. In *Advances in neural information processing systems*, pp. 1655–1663.

Naik, P. A., M. K. Mantrala, and A. G. Sawyer (1998). Planning media schedules in the presence of dynamic advertising quality. *Marketing science 17*(3), 214–235.

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.

Pollett, P. (2003). Integrals for continuous-time markov chains. *Mathematical biosciences 182*(2), 213–225.

Puri, P. S. (1966). On the homogeneous birth-and-death process and its integral. *Biometrika 53*(1-2), 61–71.

Ray, W. and F. Margo (1976). The inverse problem in reducible markov chains. *Journal of Applied Probability*, 49–56.

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

Talluri, K. T. and G. J. Van Ryzin (2006). *The theory and practice of revenue management*, Volume 68. Springer Science & Business Media.

Truzzi, F. S., V. F. d. Silva, A. H. R. Costa, and F. G. Cozman (2012). Markov decision processes for ad network optimization. In *Brazilian Conference on Intelligent Systems-BRACIS*. SBC.

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

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

Yaveroglu, I. and N. Donthu (2008). Advertising repetition and placement issues in on-line environments. *Journal of Advertising 37*(2), 31–44.

CHAPTER 3:

Competitive Real-Time Policies for the Allocation of Online Guaranteed Targeted Display Advertising

3.1 Introduction

In this paper we turn our focus to the planning of online guaranteed targeted display advertising. A *guaranteed* contract typically reserves in advance a certain number of ad impressions to be shown in certain slots on specific pages of the publisher's website over a certain time period. A *targeted* campaign further requires the ad to be shown only to users of certain demographic groups (e.g. age, gender, income level, location) and/or behavioral attributes (e.g. shopping).The publisher is paid either based on impressions delivered (CPM), number of clicks (CPC), or a conversion/purchase on advertiser's website (CPA). User arrivals, in aggregate, follow certain patterns which enables the publisher to forecast the supply of impressions and sell guaranteed advertising campaigns well in advance. Over short time intervals, however, the arrival of each user type is a lot less predictable. Given the multi-billion dollar revenue that large publishers such as Google and Facebook earn annually, a few percent improvement in drawing the *correct* ad for each slot on the web page that each user sees can improve publisher revenues by tens of millions of dollars, increase advertising efficiency and return on investment for advertisers, and enhance user experience.

We attempt to compare the performance of offline solution techniques against online

policies for the allocation of guaranteed display advertising in online media. We test a variety of scaling functions to find the best online policy. We show that a policy that assigns each impression to the open campaign with highest scaled penalty $c_k e^{-\tilde{d}_k}$, in which $\tilde{d}_k$ denotes the fraction of the campaign k served thus far, happens to have a worst-case competitive ratio of 50% with a potential of improving to near 80% as the campaigns get close to being fully served.

We consider two benchmark offline models: (1) a linear program that minimizes under-delivery penalty, and (2) a stochastic program with chance constraints that maximizes the (weighted) probability that campaigns are fully satisfied. Stochastic programming formulation of the online ad planning problem is also new to the literature and is developed in this paper for the sake of having a non–deterministic yet offline planning model for benchmark. Our results show that even with moderately noisy supply forecasts (MAPE of 25%), the online policy can outperform an offline linear programming solution. Moreover, the simple online policy can outperform solutions obtained using offline stochastic programming, even when the supply realizations are drawn from the same distribution modeled in the stochastic program.

The rest of the paper is organized as follows: In §3.2, we provide a review of existing literature, classified into deterministic models (§3.2.1), stochastic models (§3.2.2), and online matching algorithms (§3.2.3). Then we provide our numerical experiments in §3.3. Concluding remarks appear in the final §3.4.

3.2 Review of Existing Literature

Modeling of the ad allocation problem as a *transportation problem* (i.e., bipartite graph), with supply and demand nodes that respectively represent viewer types (user demographics) and ad campaigns (contracts), has been a very common and useful modeling approach and quite successful in practice. This representation was discussed in much detail in Chapter 1 (see §1.3.1). Each partition of user impressions (e.g. based on website, position of ad on the

webpage, user demographics and behavioral attributes) is modeled as a *supply node*, indexed by $i \in \mathcal{I} \equiv \{1, \ldots, M\}$, and each ad campaign/contract is modeled as a *demand node*, indexed by $k \in \mathcal{K} \equiv \{1, \ldots, N\}$ on the right. The *arcs* $(i, k) \in \mathcal{T}$ represent the targeting criteria of the campaigns, i.e., which impressions are eligible to be served with ads from which campaigns. We use $\Gamma(\cdot)$ notation for node adjacency list on the graph. That is, $\Gamma(k) = \{i : (i, k) \in \mathcal{T}\}$ denotes the set of all impressions i eligible for contract k, and $\Gamma(i) = \{j : (i, k) \in \mathcal{T}\}$ denotes the set of all eligible contracts k that can be delivered to an impression of type i. Let s_i denote the expected supply of impressions from each supply node (user partition) i over the planning horizon, let d_k denote the total number of impressions that are reserved by (guaranteed to) campaign j across users of type $\Gamma(k)$.

The publisher's problem is to find the optimal fraction of impressions i that should be allocated to each contract k, denoted x_{ik}, so as to maximize/minimize a particular objective function. Given the supply forecasts s_i, a large-scale math program is solved by the publisher to determine the best allocation plan, x^*_{ik}, over some time horizon in near future. This is referred to as *offline planning*. A typical real-life graph can have millions of supply nodes and hundreds of thousands of demand nodes. Therefore, even if the math program is formulated as a linear program (LP), it often requires special algorithmic treatment to be solved within time and memory limitations. During the serving horizon, as users make visits to publisher's webpage, the optimal fractional solution x^*_{ik} is treated as the *probability* that each campaign $k \in \Gamma(i)$ should be (randomly) drawn for a type-i user.

In the following two sections we review two offline planning approaches: one that uses expected (deterministic) supply forecasts (§3.2.1), and one that employs (stochastic) distributional forecasts (§3.2.2). Then, in §3.2.3, we turn to a separate line of research that employs online matching heuristics and requires no supply forecast and no offline planning problem to be solved.

3.2.1 Deterministic Offline Models

Deterministic mathematical models for allocation of impression-based campaigns were reviewed in Chapter 1 (see §1.2). For Cost-per-Click (CPC) advertising, the common objective is to maximize the total click-through rate: $\sum_{(i,k)\in\mathcal{T}} c_{ik}s_i x_{ik}$ or the publisher's expected revenue: $\sum_{(i,k)\in\mathcal{T}} r_{ik}c_{ik}s_i x_{ik}$, where c_{ik} denotes the probability that a viewer of type i would click on ad k, and r_{ik} is the transaction revenue for publisher if a click occurs. For Cost-per-Impression (CPM) guaranteed advertising, a typical objective function includes an under-delivery penalty, as well as a representativeness (fairness) measure which captures how uniformly the delivery of each campaign is spread across its eligible supply $\Gamma(i)$. This prevents the publisher from satisfying a campaign using only a small subset of targeted demographics. Of these models, Bharadwaj et al. (2012) was thoroughly analyzed in Chapter 1 (see §1.3.1). Without a representativeness metric, the math program takes the form of a linear program:

$$\text{Minimize:} \quad \sum_k c_k u_k \tag{3.1a}$$

$$\text{s.t.} \quad \sum_{i\in\Gamma(k)} s_i x_{ik} + u_k \geq d_k \quad \forall k \tag{3.1b}$$

$$\sum_{k\in\Gamma(i)} x_{ik} \leq 1 \quad \forall i \tag{3.1c}$$

$$x_{ik},\ u_k \geq 0 \quad \forall i,k \tag{3.1d}$$

Demand constraint (3.1b) requires the total number of impressions allocated to each contract k to exceed its demand d_k, or otherwise we have an under-delivery of u_k impressions. Supply constraint (3.1c) implies that we cannot allocate more than 100% of supply from each node i. The objective function (3.1a) penalizes under-delivery where each contract has an under-delivery penalty c_k per impression. After a substitution $u_k = d_k - \sum_{i\in\Gamma(k)} s_i x_{ik}$ and removing the resulting constant $\sum_k c_k d_k$ from the objective function, the above math program can be

writtes as:

$$\text{(LP)} \qquad \text{Maximize:} \quad \sum_{(i,k)\in\mathcal{T}} c_k s_i x_{ik} \qquad \underline{\text{Duals:}} \tag{3.2a}$$

$$\text{s.t.} \quad \sum_{i\in\Gamma(k)} s_i x_{ik} \le d_k \quad \forall k \qquad \alpha_k \tag{3.2b}$$

$$\sum_{k\in\Gamma(i)} x_{ik} \le 1 \quad \forall i \qquad \beta_i \tag{3.2c}$$

$$x_{ik} \ge 0 \quad \forall i,k \tag{3.2d}$$

which is a (weighted) maximum flow problem on a bipartite graph. The new constraint (3.2b) corresponds to $u_k \ge 0$ in (3.1) and resembles a budget constraint, as opposed to a demand constraint.

3.2.2 Stochastic Offline Models

Cholette et al. (2012) studies the planing of non-guaranteed (auction-type) advertising under probabilistic budget constraints. To the best of my knowledge, there is no paper that considers the planning of *guaranteed* display ads (i.e., one that incorporates demand constraints and aims for minimal under-delivery and/or maximal representativeness) with chance constraints or recourse decisions. To serve as a benchmark, in this section we develop an offline model which produces an impression allocation plan using probabilistic/distributional information on the supply of different demographics.

In the context of GTDA planning, the demand parameters d_k are deterministic and given. The realized supply vector s_i, however, is random and in deterministic formulations such as (3.2) is replaced by its expected value (forecast). The fractional decision variables, x^*_{ik}, ensure that the supply constraint is never violated, because allocation of the supply of each user-type i is determined as fractions of the to-be-realized supply, and (3.2s) ensures we do not allocate more than 100%. However, the actual volume of impressions delivered to each campaign k, i.e., $\sum_{i\in\Gamma(k)} s_i x_{ik}$, does depend on the realization of s_i values.

Assume each s_i is a random variable with mean μ_i and variance σ_i^2, and let $\sigma_{ii'}$ denote the covariance between each pair of supply nodes i and i'. We consider the following chance-constrained program:

$$\text{Maximize:} \qquad \prod_j (\eta_k)^{c_k} \equiv \sum_k c_k \log \eta_k \tag{3.3a}$$

$$\text{s.t.} \qquad P\Big(\sum_{i \in \Gamma(k)} s_i x_{ik} \geq d_k \Big) \geq \eta_k \qquad \forall j \tag{3.3b}$$

$$\sum_{k \in \Gamma(i)} x_{ik} \leq 1 \qquad \forall i \tag{3.3c}$$

$$0 \leq \eta_k \leq 1 \qquad \forall j \tag{3.3d}$$

$$x_{ik} \geq 0 \qquad \forall (i,k) \in \mathcal{T} \tag{3.3e}$$

Demand constraints (3.3b) require each campaign k to be fully satisfied with a probability of at least η_k, which is also formulated as a decision variable. The objective is to maximize the satisfiability probabilities η_k in a weighted scheme which resembles a maximum likelihood metric[1].

The volume of supply delivered to each campaign k will be a random variable $\bar{\xi}_k = \sum_{i \in \Gamma(k)} s_i x_{ik}$ with mean $\bar{\mu}_k = \sum_{i \in \Gamma(k)} x_{ik} \mu_i$ and variance $\bar{\sigma}_k^2 = \sum_{i,i' \in \Gamma(k)} x_{ik} x_{i'k} \sigma_{ii'}$. Let $\xi_k = (\bar{\xi}_k - \bar{\mu}_k)/\bar{\sigma}_k$ denote the corresponding *standardized* random variable with a mean of zero and a standard deviation of 1. Let $F_{\xi k}^{-1}(\eta_k)$ denote the inverse cumulative probability function of ξ_k. Following a change of variable $\hat{\eta}_k = -\log \eta_k$, derived in Appendix 3.A, we can

[1] After a number of numerical tests, we found that other reasonable, yet simple, objective functions such as Maximizing $\sum_k c_k \eta_k$ or Minimizing $\sum_k \eta_k / c_k$ result in poor convergence, when the corresponding deterministic-equivalent math program is solved using `IPOPT` solver in AMPL. At the same time, these alternatives performed no better than (3.3a) on most test instances.

transform (3.3) into a convex deterministic-equivalent form:

$$\text{(SP)} \qquad \text{Minimize:} \quad \sum_{k} c_k \hat{\eta}_k \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{i \in \Gamma(k)} \mu_i x_{ik} + F_{\xi k}^{-1}\big(1 - e^{-\hat{\eta}_j}\big) y_k \;\geq\; d_k \qquad \forall k \tag{3.4b}$$

$$y_k^2 \;=\; \sum_{i,i' \in \Gamma(k)} \sigma_{ii'} x_{ik} x_{i'k} \qquad \forall k \tag{3.4c}$$

$$\sum_{k \in \Gamma(i)} x_{ik} \leq 1 \qquad \forall i \tag{3.4d}$$

$$x_{ik} \geq 0,\; y_k \geq 0,\; \hat{\eta}_k \geq 0 \qquad \forall (i,k) \in \mathcal{T} \tag{3.4e}$$

which has a linear objective function, a set of non-linear constraints (3.4b), conic constraints (3.4c), and linear constraints (3.4d)[2]. Without any algorithmic / large-scale treatment of this formulation, we use the general interior-point solver `IPOPT` for AMPL to solve (3.4). We assume ξ_k are standard normal random variables, i.e., $F^{-1}(\cdot)$ is the inverse cumulative distribution function of a standard normal random variable, which was accessible as "`gsl_cdf_ugaussian_Pinv(p)`" through AMPL's GNU Scientific Library[3].

3.2.3 Online Algorithms

In all modeling approaches presented in the previous sections, the publisher needs to: (1) create a supply forecast for a certain horizon in future, (2) solve a large-scale optimization problem prior to the serving period (offline phase), and (3) use the static optimal solution, treated as ad serving probabilities, to assign ad impressions to contracts upon each arrival throughout the serving horizon (online phase). This approach requires the publisher to solve a large-scale optimization problem frequently throughout the day so the solution is adapted to the most recent supply forecasts and the campaigns' progress status. Even by employing specialized optimization algorithms, such as the SHALE by Bharadwaj et al. (2012), it can

[2]The reader may refer to Prékopa (2013) for standard techniques in translating stochastic programs with chance constraints into equivalent deterministic form.

[3]Available for download at: http://ampl.com/resources/extended-function-library/

take multiple hours to solve the offline math program, and the resulting policy is prone to mistakes since the solution remains stationary until the next re-solving period.

There is a separate stream of research that eliminates both the need for supply forecasting and the need for solving an offline planning problem. These approximate/myopic heuristics use minimal (simplest) state information (such as campaigns' progress status) to select an eligible ad upon each user visit. Mehta (2012) provides the most extensive review of this literature. We provide a short excerpt below for quick reference. Buchbinder and Naor (2009) also provides a very instructive chapter on how a primal-dual analysis can be used for designing online algorithms in a variety of problem settings such as set covering, load balancing, routing, ad auction revenue, etc.

Online algorithms are designed to deal with *online input*, which is unknown in advance and revealed incrementally at the same time that the algorithm has to make decisions. I limit the scope of this study to the case of *adversarial* input. That is, there is absolutely no prior knowledge on the size (supply forecast) or type (graph connectivity) of the input. Therefore, a bound on the performance of the online algorithm should consider the worst-case input. Five classes of problems have been studies for the allocation of online advertising:

1. **Online bipartite matching:** There is a graph $G(\mathcal{I}, \mathcal{K}, \mathcal{T})$, of which one side, $\mathcal{K}$ (campaign list), is known in advance, and the other side, $\mathcal{I}$ (user types) along with connectivity $\mathcal{T}$ arrives online (one impression at a time). The goal is to maximize the number of matchings.

$$\text{Maximize} \quad \left\{ \sum_{i,k} x_{ik} \;\middle|\; \sum_{i} x_{ik} \leq 1,\ \forall k, \quad \sum_{k} x_{ik} \leq 1,\ \forall i, \quad x_{ik} \in \{0,1\} \right\}$$

   This problem, along with the *optimal* online policy, has been proposed by Karp et al. (1990). It is optimal to create a random permutation of the known vertices $\mathcal{K}$ beforehand, and then match each arriving node $i \in \mathcal{I}$ with the first available node $k \in \Gamma(i)$ in the permutation. In the worst case, this policy performs within $1 - 1/e \simeq 0.63$ of

optimality (if we knew the entire graph beforehand). Interestingly, this approach (of randomly sorting nodes only once beforehand) differs from a policy that matches each vertex $i \in \mathcal{I}$ to one of the available $k \in \Gamma(i)$ at random (i.e., a random number is drawn upon each arrival). This latter has a competitive ratio of 0.5. A deterministic algorithm that matches an arrival to *any* available campaign also has a competitive ratio of 0.5 (see Mehta 2012, p.287).

2. **Online vertex-weighted bipartite matching:** A generalization of online bipartite matching in which each vertex (campaign) $k \in \mathcal{K}$ has a non-negative weight c_k, and the goal is to maximize the sum of weight of vertices in J that are matched.

$$\text{Maximize} \quad \left\{ \sum_{i,k} c_k x_{ik} \;\middle|\; \sum_i x_{ik} \leq 1, \; \forall k, \quad \sum_k x_{ik} \leq 1, \; \forall i, \quad x_{ik} \in \{0,1\} \right\}$$

   The *optimal* online policy, due to Aggarwal et al. (2011), is to create for each (known) vertex $k \in \mathcal{K}$ an *adjusted* weight $\hat{c}_k = c_k(1 - e^{r_k - 1})$ prior to the serving time, where $r_k \sim U[0,1]$ is a uniform random variable. Then we match each arrival $i \in \mathcal{I}$ to the available $k \in \Gamma(i)$ with maximum $\hat{c}_k$. The worst-case performance is $1 - 1/e \simeq 0.63$ of optimality.

3. **Adwords problem:** Each vertex (campaign) $k \in \mathcal{K}$ has a budget B_k, and edges $e \in (i,k)$ have bids b_{ik} (denoting how much advertiser k values a user type i). When we match an arriving vertex $i \in \mathcal{I}$ to a neighbor $k \in \Gamma(i)$, the budget B_k depletes by b_{ik}. When a vertex (campaign) depletes its entire budget, then it becomes unavailable. The goal is to maximize the total budget spent (revenue of the publisher). This is exactly the problem setting for auction-based non-guaranteed ad planning.

$$\text{Maximize} \quad \left\{ \sum_{i,k} b_{ik} x_{ik} \;\middle|\; \sum_i b_{ik} x_{ik} \leq B_k, \; \forall k, \quad \sum_k x_{ik} \leq 1, \; \forall i, \quad x_{ik} \in \{0,1\} \right\}$$

   The best known online policy, which has a competitive ratio of $1 - 1/e \simeq 0.63$, is due to Mehta et al. (2007). The idea is to match each arrival $i \in \mathcal{I}$ to the available $k \in \Gamma(i)$ with

maximum *scaled* bid $\hat{b}_{ik} = b_{ik}\left(\frac{1-e^{\tilde{B}_k-1}}{1-e^{-1}}\right)$ where $\tilde{B}_k$ is the fraction of the budget spent so far. Note that $\hat{b}_{ik}$ decreases from b_{ik} to 0 as the budget depletes ($\tilde{B}_k$ approaches 0). Buchbinder et al. (2007) provide a clever proof of this bound using primal-dual analysis and develop some extensions. These bounds, however, assume that the bid-to-budget ratio tends to zero.

4. **Display Advertising:** Each vertex (campaign) $k \in \mathcal{K}$ has an integral capacity d_k (demand), which is an upper bound on how many vertices (impressions) $i \in \mathcal{I}$ can be matched to k. Each edges $(i,k) \in \mathcal{T}$ has a weight c_{ik} (the quality of user i for advertiser k) . The goal is to maximize the total weight of edges matched (total quality of serving).

$$\text{Maximize} \quad \left\{ \sum_{i,k} c_{ik}x_{ik} \;\middle|\; \sum_{i} x_{ik} \leq d_k,\ \forall k, \quad \sum_{k} x_{ik} \leq 1,\ \forall i, \quad x_{ik} \in \{0,1\} \right\}$$

Note that when $c_{ik} = c_k$ for all k, i.e., the weights do not depend on i, the problem becomes an instance of the Adwords problem (to see this, multiply both sides of each demand constraint by c_k, and defined budget as $B_k = c_k d_k$). However, for the general case of c_{ik} weights, it is not possible to derive any non-trivial competitive ratio: If the input is adversarial, the adversary can sort arrivals so that we observe smallest c_{ik} until budgets are depleted, and then we observe arrivals with infinitely large c_{ik}. A side-step (which makes the setting less applicable to the practice of online advertising) has been to assume a *free disposal* property: a vertex (campaign) $k \in \mathcal{K}$ is allowed to be matched more times than its capacity d_k, but the gain is evaluated based on the d_k highest weight edges matched. With this assumption, Feldman et al. (2009) design an online algorithm which uses bid-scaling and achieves a competitive ratio of $1 - 1/e \simeq 0.63$ as capacities $d_k \to \infty$.

5. **Generalized Assignment Problem (GAP):** This is a generalization of all the problems above. Each vertex (campaign) $k \in \mathcal{K}$ has a budget, each matching $(i,k) \in \mathcal{T}$

involves a bid b_{ik} that depletes the budget B_k, and it provides a certain matching quality c_{ik} which is being maximized:

$$\text{Maximize} \quad \left\{ \sum_{i,k} c_{ik} x_{ik} \;\middle|\; \sum_{i} b_{ik} x_{ik} \leq B_k, \; \forall k, \quad \sum_{k} x_{ik} \leq 1, \; \forall i, \quad x_{ik} \in \{0,1\} \right\}$$

The approximability of this problem has been studied by Chakrabarty and Goel (2010), and is beyond the scope of this paper.

3.3 Numerical Experiments

The worst-case performance of online algorithms usually arises in graphs with few nodes, limited connectivity, and more importantly, small budget or capacity (which translates each "mistake" into a sizable loss of optimality). In real-life instances of graphs that arise in online advertising, especially in the planning of guaranteed display ads, the s_i and d_k values are very large (hundreds of thousands or millions). Therefore, the combinatorial difficulty of the online allocation is not extreme. Therefore, such simple algorithms can performs quite acceptably. The question we follow in the rest of this paper is whether (and when) online algorithms can outperform offline linear or stochastic programming models, presented in §3.2.1 and §3.2.2.

We start by examining a few *bid-scaling functions* for the online heuristic in §3.3.1. It happens that a particular functional choice, not examined in the literature before, outperforms others on our randomly constructed instances. We derive a competitive ratio for the performance of this scaling function. Then, in §3.3.2 we test how much forecast noise is enough to make the offline (LP) from §3.2.1 worse than the simple online heuristic. Finally, in §3.3.3 we compare the performance of our offline stochastic program (SP) from §3.2.2 against the (LP) and the online algorithms when the supply is drawn from a known (joint) distribution.

In all our numerical tests that follow, we synthetically generate instances of the bipartite graph under the following considerations to ensure that the graph structure and parameters

are reasonable:

Graph Structure: All graphs contain 50 supply nodes and 20 demand nodes. Each campaign is connected to a certain number of supply nodes, chosen uniformly at random. At random, about 20% of demand nodes are high-targeting (connected to $\sim$ 50% of supply nodes at random), 30% are moderately targeting (connected to $\sim$ 15% of supply nodes), and the other 50% are low-targeting (connected to $\sim$ 5% of supply nodes).

Supply Mean and Covariance: The mean value of each supply node, μ_i, is drawn randomly from an exponential distribution with mean parameter 1000. Then, each supply node is assigned a coefficient of variation, C_i, drawn uniformly from the interval $[0.2, 1.0]$. The standard deviations are then given by $\sigma_i = C_i\mu_i$. The coefficient of correlation between each pair of supply $\rho_{ii'}$ is taken from the interval $[-0.5, 1]$ proportional to the cardinality of the set $\Gamma(i) \cap \Gamma(i')$, i.e., the number of campaigns that target both i and i'. Sharing too many (resp., too few) campaigns suggests that the two supply nodes share many (resp., very few) common attributes and therefore their supply should be positively (resp., negatively) correlated. The covariance matrix is then given by $\sigma_{ii'} = \rho_{ii'}\sigma_i\sigma_{i'}$. To induce positive-definiteness, an existing Matlab code[4] was run to find the nearest positive semi-definite matrix to the one produced with the above approach.

Demand Parameters: The demand for each campaign was assigned by applying a similar algorithm to the high water mark (HWM) method, proposed by Bharadwaj et al. (2012). A value of $\theta_k \in [0.1, 0.5]$ was assigned to each demand node uniformly at random. Then, in a random order, each campaign was assigned a θ_k proportion (or whatever leftover, if less than θ_k) from all supply nodes $i \in \Gamma(k)$. The final allocation was then scaled so that the network sellthrough (defined as ratio between aggregate demand to aggregate supply: $\sum_k d_k / \sum_i s_i$) was set close to 100%. My observation was tat a low sellthrough makes the problem too easy for the online algorithm (so all offline and online algorithms perform optimality). Also, with

[4] `nearestSPD(.)` by John D'Errico, which follows the derivation of Higham (1988).

a sellthroughs much higher than 100% a significant under-delivery is unavoidable, and again all offline and online algorithms perform similarly. A sellthrough near 100% seemed to bring out the most contrast among the performances of different algorithms.

Penalty Weights: The under-delivery weights c_k were assigned randomly from integer between 1–4.

3.3.1 Choice of Scaling Function for Online Algorithm

We consider the following problem for our online algorithm, involving only under-delivery penalty:

$$\text{(OP)} \quad \text{Maximize:} \quad \sum_{i,j} c_k x_{ik} \qquad \underline{\text{Duals:}} \tag{3.5a}$$

$$\text{s.t.} \quad \sum_{i} x_{ik} \leq d_k \qquad \forall k \qquad \alpha_k \tag{3.5b}$$

$$\sum_{k \in \Gamma(i)} x_{ik} \leq 1 \qquad \forall i \qquad \beta_i \tag{3.5c}$$

$$x_{ik} \in \{0,1\} \qquad \forall (i,k) \tag{3.5d}$$

Note that (OP) is quite similar to the (LP) formulation (3.2) except we do not use any supply forecast s_i. Instead, each supply node i represents a single impression. Note that we cannot solve (OP) and obtain an offline solution beforehand since we do not have any information about the arrivals or the connectivity in the graph prior to actually observing them. We use (OP) only to derive an online policy (using duality theory) and to test the performance of the online heuristic after all supply is realized (ex-post optimal solution). The dual problem to the LP-relaxation of (OP) is given by:

$$\text{(OD)} \quad \text{Minimize:} \quad \sum_{k} d_k \alpha_k + \sum_{i} \beta_i \qquad \underline{\text{Duals:}} \tag{3.6a}$$

$$\text{s.t.} \quad \alpha_k + \beta_i \geq c_k \qquad \forall (i,k) \qquad x_{ik} \tag{3.6b}$$

$$\alpha_k \geq 0, \ \beta_i \geq 0 \tag{3.6c}$$

The following observations can be made:

1. Constraints (3.6b) can be rearranged as: $\beta_i \geq c_k - \alpha_k,\ \forall k \in \Gamma(i)$. Since β_i are minimized in the objective, we should have: $\beta_i^* = \max_{k\in\Gamma(i)}\{c_k - \alpha_k^*\}$, i.e., if we knew the optimal α_k^*, the optimal β_i^* could be found as such.

2. Complementary slackness condition for (3.6b) implies that $x_{ik}^* = 0$ for any (i,k) for which is (3.6b) is non-binding. That is, only $k^* = \arg\max_{k\in\Gamma(i)}\{c_k - \alpha_k\}$, which makes (3.6b) bind, allows $x_{ik*} = 1$. Therefore, the i'th page visit should be served to k^* derived above.

3. Complementary slackness condition for (3.5b) implies $\alpha_{k_0} = 0$ for any campaign k_0: $\sum_i x_{ik_0} < d_{k_0}$ (non-binding).

4. The non-negativity constraint $\beta_i \geq 0$ together with (3.6b) implies that $\alpha_k \in [0, c_k]$.

5. After t arrivals, if some campaign is fully satisfied, $\sum_{i\leq t} x_{ik_0} = d_{k_0}$, then we should no longer serve to k_0. That is, we should set: $\alpha_{k_0} = c_{k_0}$.

In summary, upon each arrival, we should serve the user i using the campaign with maximum $c_k - \alpha_k$. The value of α_k should start from zero and reach c_k as soon as the campaign k is fully satisfied.

Let $\tilde{d}_k \in [0,1]$ denote the fraction of the campaign demand d_k which is served so far. Let $\alpha_k = c_k\phi(\tilde{d}_k)$ where $\phi(\cdot)$ is a monotonic/increasing function over $[0,1] \to [0,1]$. The following six functional forms all have the correct property of leading α_k from zero to c_k as $\tilde{d}_k$ goes from zero to one:

(0) Greedy: $\phi_0(\tilde{d}_k) = \lfloor \tilde{d}_k \rfloor$,

(1) $\phi_1(\tilde{d}_k) = \tilde{d}_k$,

(2) $\phi_2(\tilde{d}_k) = 1 - e^{-\tilde{d}_k}$,

(3) $\phi_3(\tilde{d}_k) = \frac{1-e^{-\tilde{d}_k}}{1-e^{-1}}$,

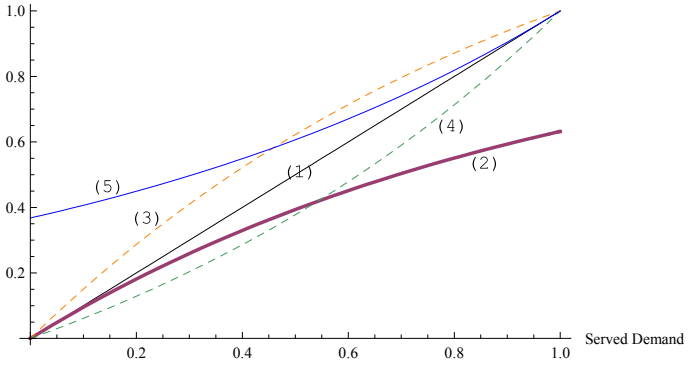


Figure 3.1: Scaling Functions $\phi(\tilde{d}_k)$ Tested for the Online Algorithm

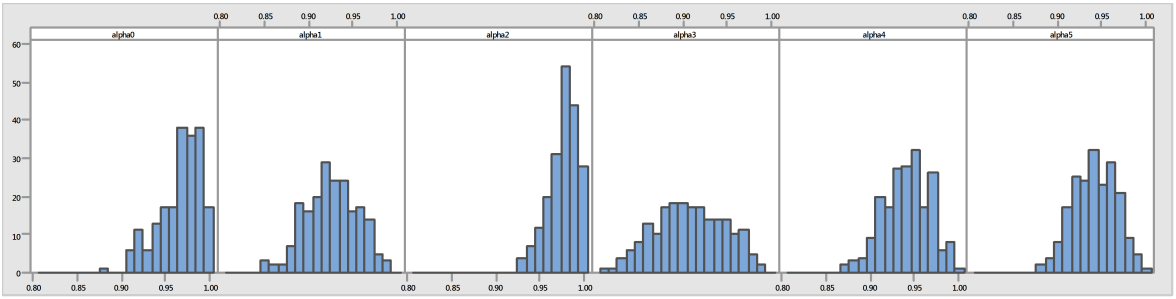


Figure 3.2: Performance of Difference Scaling Functions $\phi(\tilde{d}_k)$.

(4) $\phi_4(\tilde{d}_k) = \frac{e^{\tilde{d}_k}-1}{e-1}$,

(5) $\phi_5(\tilde{d}_k) = e^{\tilde{d}_k-1}$.

The above functions are plotted on Figure 3.1. Note that the choice of $\phi_4(\tilde{d}_k)$ results in the policy that was studied by Mehta et al. (2007) with a competitive ratio of $1-1/e$.

We tested the performance of the online algorithm, using each of the above scaling functions over 200 randomly-generated graphs. Figure 3.2 shows the histogram of results for each scaling function. The horizontal axis shows the optimal value in relation to the ex-post optimal (where 1 means the online algorithm attained the ex-post optimal value). The vertical axis shows the frequency of instances that resulted in a specific performance.

The results suggest that the functional choice $\phi_2(\tilde{d}_k) = 1-e^{-\tilde{d}_k}$ provides the best solution most often, followed by the greedy choice of $\phi_0(\tilde{d}_k) = \lfloor \tilde{d}_k \rfloor$. Note that $\phi_0(\cdot)$ results in a policy that simply gives each impression i to the open campaign $k \in \Gamma(i)$ with the highest penalty

c_k. It is shown that the greedy serving policy has a competitive ratio of 0.5 (Mehta 2012, p.323). We use a similar technique to obtain the competitive ratio of $\phi_2(\cdot)$ which assigns the impression to the open campaign with highest *scaled* penalty $c_k e^{-\tilde{d}_k}$.

The idea of the proof is showing that, regardless of the arrival stream, the primal objective value (3.5a) produce by the online policy is greater than a ρ-fraction of the dual objective value (3.6a). Then, we have:

$$(OP)_{ALG} \geq \rho(OD)_{ALG} \geq \rho(OD)^* \geq \rho(OP)^*_{LP} \geq \rho(OP)^*_{BIN} \qquad \Rightarrow \qquad \frac{(OP)^*_{ALG}}{(OP)^*_{BIN}} \geq \rho \quad (3.7)$$

The first inequality is derived by observing and tracking the increments in each of the two objective as a result of online policy, and is specific to the problem and the online allocation rule. The second inequality follows from the fact that the algorithm produces only a feasible (i.e., sub-optimal) solution to (OD), which is a minimization problem. Therefore, $(OD)^* \leq (OD)_{ALG}$. The third inequality results from weak duality. For any maximization program (OP), the optimal value of the (minimization) dual program gives a upper-bound on the optimal value of the primal math program. The last inequality follows from the fact that an LP-relaxation of (OP) can produce a higher objective value than (OP) when the variables are restricted to $\{0, 1\}$ values. These comparisons show that ρ is a guaranteed performance ratio for any instance of the problem.

Theorem 8. *The scaled online algorithm that uses ϕ_2 has a competitive ratio between $1/2$ and $1/(2 - 1/e)$ under large demand assumption $d_k \to \infty$.*

Proof. It is clear that, by construction, primal and dual feasibility are never violated by the online algorithm. Suppose a $\tilde{d}_k$ fraction of campaign k (i.e., $\tilde{d}_k d_k$ impressions) has been delivered by the end of the arrival process. The primal objective in (P) must have increased by exactly $\Delta_k(OP) = \tilde{d}_k d_k c_k$ and the dual objective in (OD) must have changed by: $\Delta_k(OD) = d_k \alpha_k + \sum_{i=1}^{\tilde{d}_k d_k} \beta_i$.

(1) If $\tilde{d}_k < 1$:

$$\Delta_k(OD) = d_k c_k(1-e^{-\tilde{d}_k}) + \sum_{t=1}^{\tilde{d}_k d_k} c_k e^{-t/d_k} \;\simeq\; d_k c_k(1-e^{-\tilde{d}_k}) + \int_{t=0}^{\tilde{d}_k} c_k e^{-t}dt \;=\; 2d_k c_k(1-e^{-\tilde{d}_k})$$

$$\Rightarrow \qquad \frac{\Delta_k(P)_{ALG}}{\Delta_k(D)_{ALG}} \;=\; \frac{\tilde{d}_k}{2(1-e^{-\tilde{d}_k})} \qquad \begin{cases} \lim_{\tilde{d}_k \to 0}: & 0.5 \\ \lim_{\tilde{d}_k \to 1}: & \frac{1}{2(1-e^{-1})} \simeq 0.7909 \end{cases}$$

Note that we used the large demand assumption to convert the summation into an integral.
(2) If $\tilde{d}_k = 1$, then we manually set $\alpha_k = c_k$ so the campaign is no longer selected:

$$\Delta_k(D) = d_k c_k + \sum_{t=1}^{d_k} c_k e^{-k/d_k} \;\simeq\; d_k c_k + \int_{t=0}^{1} c_k e^{-t}dt \;=\; d_k c_k + d_k c_k(1-e^{-1}) \;=\; d_k c_k(2-e^{-1})$$

$$\Rightarrow \qquad \frac{\Delta_k(P)_{ALG}}{\Delta_k(D)_{ALG}} \;=\; \frac{1}{2-1/e} \;\simeq\; 0.6127$$

Using (3.7), the above suggest that the worst-case bound of the online policy that uses $\phi_2(\cdot)$ is at 0.5 and no better than the greedy policy. However, as more campaigns get closer to being fully satisfied, the bound improves to near 80% which is better than the best-known bound of $1-1/e \simeq 0.63$. □

As we found, numerically, that the choice of $\phi_2(\tilde{d}_k)$ and online policy: $k^* = \arg\max_{k\in\Gamma(i)}\{c_k e^{-\tilde{d}_k}\}$ outperforms others, we will use this scaling function throughout the rest of this paper.

3.3.2 Competency of Offline Models with Noisy Forecast

On a particular instance of the graph, we solve the (LP) problem from §3.2.1 using the supply forecast s_i which is based on the mean supply expected in each node. We then compare the performance of the resulting offline static solution x^*_{ik} against that of the online algorithm when the actual number of impressions $s_i^{(a)}$ in each node is drawn randomly, from a log-normal distribution[5] with mean s_i. We vary the coefficient of variation of the log-normal distribution

[5] Appendix 3.B describes how standard normal random numbers can be transformed into log-normal with desired mean and standard deviation / covariance matrix.

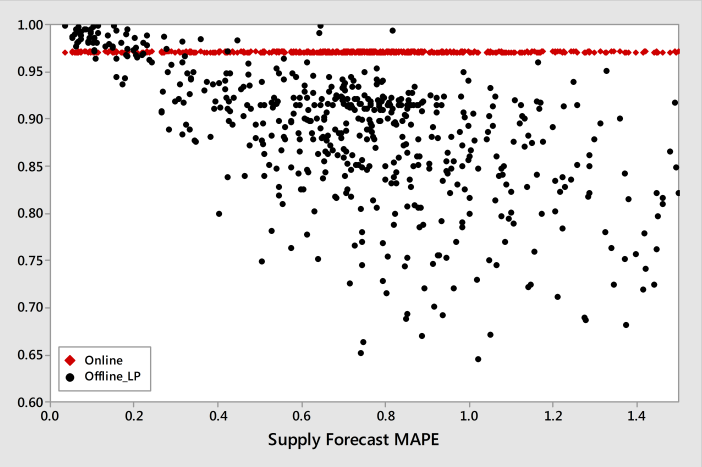


Figure 3.3: Performance of Online Policy against the (LP) under Noisy Forecast.

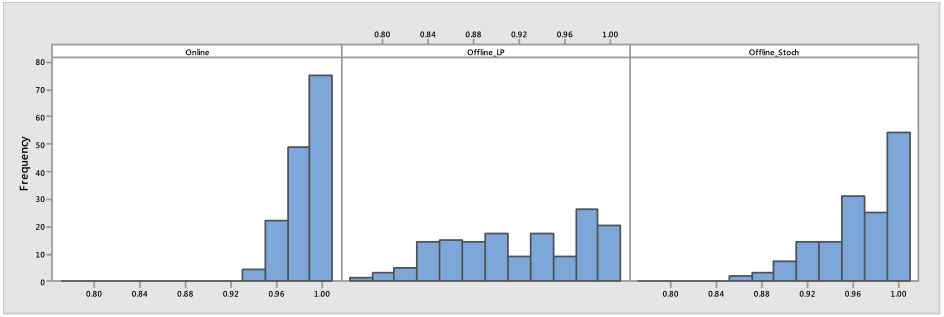
to change the (theoretical) degree of forecast noise. For each instance, the empirical degree of forecast noise is measured using Mean Absolute Percent Deviation (MAPD) measure:

$$MAPD = \frac{1}{M}\sum_{i=1}^{M}\frac{|s_i - s_i^{(a)}|}{s_i}$$

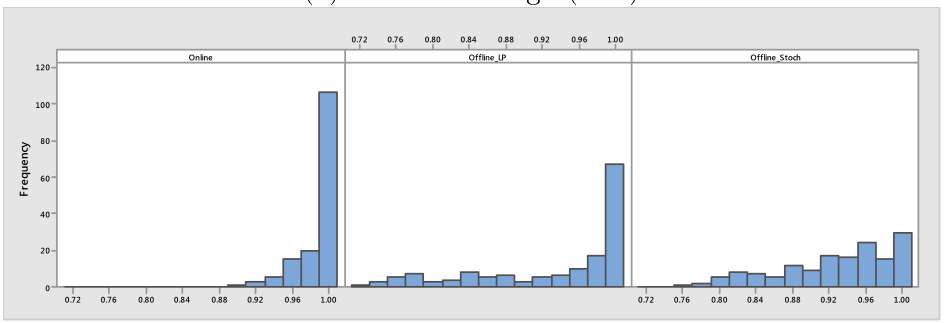
Figure 3.3 shows that for any MAPD greater than 25% (i.e., when the forecast in every supply node is beyond 25% off from the actual arrivals), the online algorithm outperforms the offline (LP) solution. Note that the online algorithm does not require or use any forecast, and its performance remains at 97% of optimality, regardless of the degree of forecast noise. This shows the importance and power of online heuristics.

3.3.3 Online Algorithm vs. Offline Stochastic Solution

Finally, we compare the performance of our online algorithm against the stochastic program (SP) introduced in §3.2.2. For this test, we created graphs with low (0.77) and high (1.52) sellthrough levels. For each graph, we sampled 150 instances of the supply vector $s_i^{(a)}$. The mean was set to the value used in (LP) and (SP), and the covariance matrix was the same as the one used in the (SP) model. Figure 3.4 shows the performance of different methods with regards to under-delivery penalty. The ex-post optimal solution was obtained by solving



(a) Low Sellthrough (0.77)

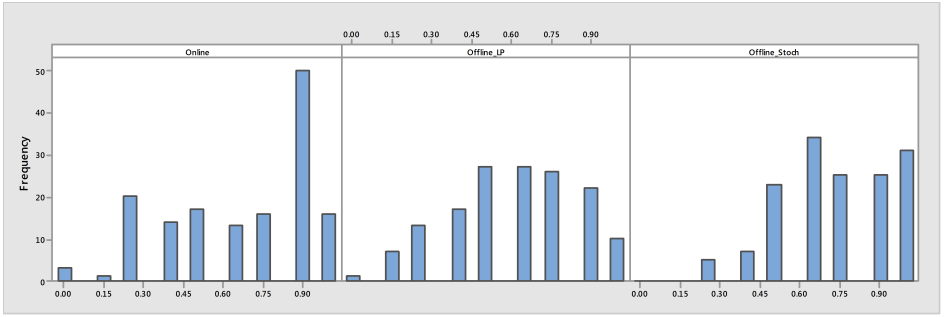


(b) High Sellthrough (1.52)

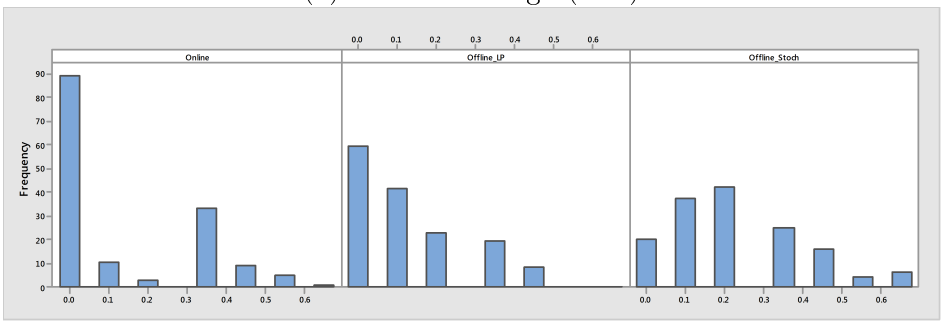
Figure 3.4: Underdelivery Penalty Performance the Online Policy vs. (LP) and (SP).

(LP) on the actual arrivals $s_i^{(a)}$. We observe that the online algorithm outperforms both offline solution, regardless of the sellthrough level. On the low-sellthrough (easy) instance, the (SP) solution outperforms the (LP) solution as it uses the supply more conservatively, with a proper anticipation of randomness in the arrivals. On the high-sellthough (difficult) instance, however, the (LP) solution performs better than (SP). This can be attributed the objective function of (SP) which aims for maximizing the probability that campaign are *fully* satisfied. When the supply is extremely short, the probability that any campaign is *fully* satisfied could tend to zero. In this case, the (SP) solution no longer recognizes / properly differentiates the campaigns based on under-delivery penalty. Therefore, (SP) solution tends to perform poorly.

To be more fair to the (SP) solution, we also measured the performance of the three algorithms on another dimension: The fraction of campaigns *fully* satisfied at the end of the horizon. Figure 3.5 shows these results. In all cases, we can see that the performance



(a) Low Sellthrough (0.77)



(b) High Sellthrough (1.52)

Figure 3.5: Fraction of Fully-satisfied Campaigns under the Online Policy vs. (LP) and (SP).

histogram is more inclined to the right under the (SP) solution. That is, the (SP) solution performs best in terms of ensuring that the highest number of campaigns are fully satisfied by the end of the horizon. This is more in line the objective function (3.4a) used in (SP).

3.4 Concluding Remarks

This paper was an attempt in comparing the performance of offline solution techniques against simple online policies for the allocation of guaranteed display advertising in online media. In particular, we considered two offline models: (1) a linear programming formulation that minimized under-delivery penalty, and (2) an offline stochastic programming formulation with chance constraints that maximized the (weighted) probability that campaigns are fully satisfied. Then we tested a variety of scaling functions to find the best online policy. We showed that a policy which assigns each impression to an unfinished campaign with highest scaled penalty $c_k e^{-\tilde{d}_k}$ (in which $\tilde{d}_k$ denotes the fraction of the campaign which has been served

thus far) can potentially have a competitive ratio of $1/2(1-1/e) \simeq 0.79$ which is better then the best known algorithm with competitive ratio $1-1/e \simeq 0.63$. Our results further showed that even with moderately noisy supply forecasts (MAPE of 25%), the online policy can outperform an offline (LP) solution. Moreover, that the simple online policy can outperform solutions obtained using offline stochastic programming, even when the supply realizations match their distributional forecast.

Bibliography

Aggarwal, G., G. Goel, C. Karande, and A. Mehta (2011). Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1253–1264. SIAM.

Bagnoli, M. and T. Bergstrom (2005). Log-concave probability and its applications. *Economic theory 26*(2), 445–469.

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. ACM.

Buchbinder, N., K. Jain, and J. S. Naor (2007). Online primal-dual algorithms for maximizing ad-auctions revenue. In *Algorithms–ESA 2007*, pp. 253–264. Springer.

Buchbinder, N. and J. Naor (2009). The design of competitive online algorithms via a primal: dual approach. *Foundations and Trends® in Theoretical Computer Science 3*(2–3), 93–263.

Chakrabarty, D. and G. Goel (2010). On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. *SIAM Journal on Computing 39*(6), 2189–2211.

Cholette, S., Ö. Özlük, and M. Parlar (2012). Optimal keyword bids in search-based advertising with stochastic ad positions. *Journal of Optimization Theory and Applications 152*(1), 225–244.

Feldman, J., N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál (2009). Online ad assignment with free disposal. In *Internet and network economics*, pp. 374–385. Springer.

Higham, N. J. (1988). Computing a nearest symmetric positive semidefinite matrix. *Linear algebra and its applications 103*, 103–118.

Karp, R. M., U. V. Vazirani, and V. V. Vazirani (1990). An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pp. 352–358. ACM.

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.

Prékopa, A. (2003). Probabilistic programming. *Handbooks in operations research and management science 10*, 267–351.

Prékopa, A. (2013). *Stochastic programming*, Volume 324. Springer Science & Business Media.

Appendices

3.A Maximizing the Probability that a Chance Constraint Holds

Consider the (univariate) random variable ξ with cumulative distribution $F(x) = P(\xi \leq x)$. Now consider the probabilistic constraint:

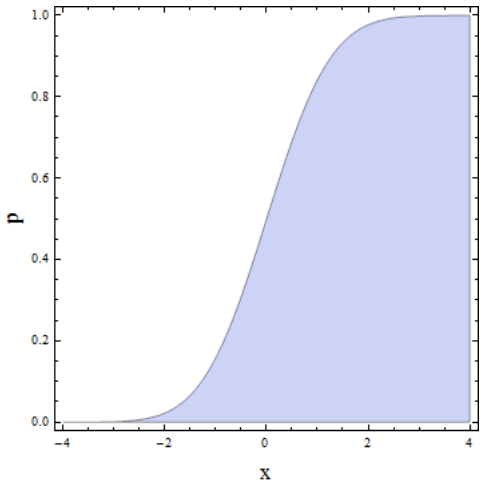
$$P(\xi \leq x) \geq p \qquad \equiv \qquad F(x) \geq p \tag{3.8}$$

If only x is a decision variable, then for any fixed parameter $p \in [0, 1]$, the constraint (3.8) is equivalent to $x \geq F^{-1}(p)$ which defines a convex feasible set for x. Similarly, if only p is a decision variable, then for any x the constraint (3.8) simply enforces an upper-bound $p \leq F(x)$. Now consider the problem in which both x and p are decision variables. The shaded area on Figure 3.6(a) shows the set of feasible (x, p) pairs that are feasible in (3.8) when ξ is a standard normal random variable. Note that this area in not convex. Therefore, the following optimization program is not convex:

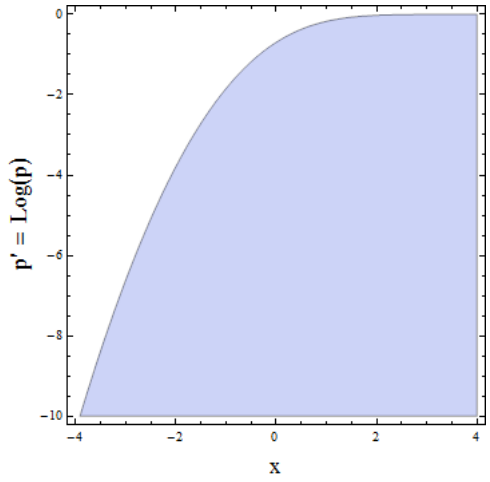
$$\begin{aligned} \text{Maximize} \quad & p \\ \text{s.t.} \quad & F(x) \geq p \quad \equiv \quad x \geq F^{-1}(p) \\ & x \in \mathcal{X},\ p \in [0, 1] \end{aligned}$$

For the case of normally-distributed ξ, note that the area in Figure 3.6(a) is convex if we further require $p \geq 0.5$ (see also Prékopa 2003, p.284). But in our particular application, this might be too restrictive and under short supply could render the problem infeasible. We show that when $F(\cdot)$ is log-concave, we can transform the constraint into a convex form for all $p \in [0, 1]$ following a simple change of variable.

If the CDF $F(\cdot)$ is log-concave, then $\log F(x)$ is a concave function of x. Therefore,



(a) $F(x) \geq p$



(b) $\log F(x) \geq \hat{p} = \log p$

Figure 3.6: Convexity of the probabilistic constraint when satisfiability probability is also a decision variable

$\log F(x) \geq \hat{p}$, i.e., super-level set of a concave function, defines a jointly convex feasible set over (x, p), as shown in Figure 3.6(b). Note that the monotonicity of the $\log(\cdot)$ function implies that constraint (3.8) is equivalent to $\log F(x) \geq \log p$, and maximizing p is equivalent to maximizing $\log p$. A change of variable $\hat{p} = -\log p$, therefore, produces the convex-equivalent program:

$$\begin{aligned} &\text{Minimize} \quad \hat{p} \\ &\quad \text{s.t.} \quad \log F(x) \geq -\hat{p} \quad \equiv \quad F(x) \geq e^{-\hat{p}} \quad \equiv \quad x \geq F^{-1}(e^{-\hat{p}}) \\ &\qquad\quad x \in \mathcal{X},\ \hat{p} \geq 0 \end{aligned}$$

More rigorously, we can show the convex transformation as follows:

$$\begin{aligned} &\text{Maximize} \quad \prod_{i=1}^{n} P(\xi_i \leq x_i), \quad \text{s.t.} \quad x \in \mathcal{X} \\ \equiv \quad &\text{Maximize} \quad \prod_{i=1}^{n} F_i(x_i), \quad \text{s.t.} \quad x \in \mathcal{X} \\ \equiv \quad &\text{Maximize} \quad \sum_{i=1}^{n} \log F_i(x_i), \quad \text{s.t.} \quad x \in \mathcal{X} \quad : \text{due to monotonicity of } \log(\cdot) \end{aligned}$$

If all $F_i(\cdot)$ are log-concave, then the objective function is concave (convex problem). Writing in epigraph form, and observing that $F(\cdot) \in [0,1] \Rightarrow \log F(\cdot) \leq 0$:

$$\equiv \quad \text{Maximize} \quad \sum_{i=1}^{n} \hat{p}_i, \quad \text{s.t. } \hat{p}_i \leq \log F_i(x_i)\ \forall i, \quad x \in \mathcal{X}, \quad \hat{p}_i \leq 0\ \forall i$$

Switching variables: $\hat{p}_i \leftarrow -\hat{p}_i$:

$$\equiv \quad \text{Minimize} \quad \sum_{i=1}^{n} \hat{p}_i, \quad \text{s.t. } -\hat{p}_i \leq \log F_i(x_i)\ \forall i, \quad x \in \mathcal{X}, \quad \hat{p}_i \geq 0\ \forall i$$

$$\equiv \quad \text{Minimize} \quad \sum_{i=1}^{n} \hat{p}_i, \quad \text{s.t. } F_i(x_i) \geq e^{-\hat{p}_i}\ \forall i, \quad x \in \mathcal{X}, \quad \hat{p}_i \geq 0\ \forall i$$

which implies $p_i := P(\xi_i \leq x_i) = F_i(x_i)$ is lower-bounded by $e^{-\hat{p}_i}$, i.e., $\hat{p}_i = -\log p_i$.

$$\equiv \quad \text{Minimize} \quad \sum_{i=1}^{n} \hat{p}_i, \quad \text{s.t. } x_i \geq F_i^{-1}(e^{-\hat{p}_i})\ \forall i, \quad x \in \mathcal{X}, \quad \hat{p}_i \geq 0\ \forall i$$

For numerical stability, it is useful to require lower- and upper-bounds on $\hat{p}_i$ variables, such as:

$$-\log(1-\epsilon) \leq \hat{p}_i \leq -\log(\varepsilon)$$

which implies $p \in [\varepsilon, 1-\varepsilon]$. Otherwise, note that as if the $\mathcal{X}$ allows $\hat{p}_i$ to approach 0 (corresponding to $p_i \to 1$), or if the solver initializes the algorithm as so, then the quantile $F^{-1}(e^{-\hat{p}_i}) \to F^{-1}(1) \to \infty$, if the support for ξ is not compact (e.g., with normal distribution). Similarly, if $\mathcal{X}$ is too restrictive and a constraint i can be satisfied with $p_i \to 0$ probability, then $\hat{p} \to \infty$ which makes the objective value unbounded (or the problem poorly scaled). As an example, $\varepsilon = 10^{-6}$ would imply $\hat{p} \in [10^{-6}, 13.82]$ which keeps the search-space very compact, yet allows $p \in [.000001, .999999] \simeq [0,1]$.

3.B Log-Normal Random Variables

In this section we summarize a few properties of the log-normal random variable which have been useful in the implementation of our numerical analysis. The log-normal PDF is not log-concave, but its CDF, which is of interest to convexity property of our optimization problem,

is log-concave (see Bagnoli and Bergstrom 2005).

Univariate distribution:

If $X \sim \log\mathcal{N}(\mu, \sigma)$, then the random variable $Y = \log X$ will be normally distributed with mean μ and standard deviation σ.

If $Y \sim \mathcal{N}(\mu, \sigma)$, then $X = e^Y$ will have a log-normal distribution with:

Mean: $m = e^{\mu+\sigma^2/2}$

Variance: $s^2 = (e^{\sigma^2} - 1)m^2$

The above imply that to generate random instances of Y with a particular mean and standard deviation (m, s), we can generate normally-distributed random numbers Y with (μ, σ) determined as follows:

$$\sigma^2 = \log\Big(\frac{s^2}{m^2} + 1\Big), \qquad \mu = \log(m) - \frac{\sigma^2}{2}$$

The exponentials of $Y \sim \mathcal{N}(\mu, \sigma)$ will have the desired log-normal distribution.

Multivariate (correlated) distribution:

If $Y = (Y_1, \ldots, Y_n) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate normal distribution, then $X = (e^{Y_1}, \ldots, e^{Y_n})$ has a multivariate log-normal distribution with:

Mean: $m_i = E[X_i] = e^{\mu_i + \sigma_{ii}/2}$

Covariance: $S_{ij} = Cov[X_i, X_j] = (e^{\sigma_{ij}} - 1)m_i m_j$

Therefore, to generate random instances of a multivariate log-normal distribution with a particular mean and covariance $(\boldsymbol{m}, \boldsymbol{S})$, we can generate multivariate normal random variables with parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ determined as follows:

$$\sigma_{ij} = \log\Big(\frac{S_{ij}}{m_i m_j} + 1\Big), \qquad \mu_i = \log(m_i) - \frac{\sigma_{ii}}{2}$$

or in matrix form:

$$\boldsymbol{\Sigma} = \log\Big(\frac{\boldsymbol{S}}{\boldsymbol{m}\boldsymbol{m}^\top} + 1\Big), \qquad \boldsymbol{\mu} = \log(\boldsymbol{m}) - \mathrm{diag}(\boldsymbol{\Sigma})/2$$

where all operations are conducted element-wise. Taking the element-wise exponentials of $Y \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ will produce the desired correlated log-normal vectors.

This is particularly useful to our simulation study, since MATLAB does not have a direct function for generating correlated multivariate log-normal random numbers. Note that it is not clear whether the $\boldsymbol{\Sigma}$ calculated from the above is positive (semi)definite, which is a property of multivariate normal distribution. It is straightforward to show that the matrix inside the $\log(\cdot)$ is positive semi-definite when $\boldsymbol{S}$ is as such. However, I could not find/prove that element-wise logarithm preserves positive definiteness. That said, I never encountered any issue with positive-definiteness of $\boldsymbol{\Sigma}$ in my test cases.