The Planning of Guaranteed Targeted Display Advertising

John Turner
The Paul Merage School of Business, University of California at Irvine, Irvine, CA 92697-3125
john.turner@uci.edu

As targeted advertising becomes prevalent in a wide variety of media vehicles, planning models become increasingly important to ad networks that need to match ads to appropriate audience segments, provide a high quality of service (meet advertisers' goals), and ensure ad serving opportunities are not wasted. We define *Guaranteed Targeted Display Advertising* (GTDA) as a class of media vehicles that include webpage banner ads, video games, electronic outdoor billboards, and the next generation of digital TV, and formulate the GTDA planning problem as a transportation problem with quadratic objective. By modeling audience uncertainty, forecast errors, and the ad server's execution of the plan, we derive sufficient conditions that state when our quadratic objective is a good surrogate for several ad delivery performance metrics. Moreover, our quadratic objective allows us to construct duality-based bounds for evaluating aggregations of the audience space, leading to two efficient algorithms for solving large problems: the first intelligently refines the audience space into successively smaller blocks, and the second uses scaling to find a feasible solution given a fixed audience space partition. Near-optimal schedules can often be produced despite significant aggregation.

Key words: guaranteed targeted display advertising, advertising, planning, aggregation

1. Introduction

Targeted ads – those shown only to audience segments requested by advertisers – are embedded in a wide spectrum of media: webpages display banner ads and video ads, video games on PC's and consoles like the XBox seamlessly integrate ads (Turner et al. 2011), and social media platforms like Facebook deliver ads that match users' profiles. Moreover, new technology is permitting targeted advertising in previously broadcast-only media: e.g., electronic billboards in some malls use eye-tracking technology to count ad views (Mandel 2007), and perhaps most importantly, the six largest cable companies in the United States are collaborating to bring targeted advertising to digital television (Arango 2008). Indeed, the market for targeted advertising is growing faster than online

advertising, which by itself is expected to nearly double by 2013 from the $23.4B (8.7% of all U.S. advertising) recorded in 2008 (cf. Interactive Advertising Bureau 2009 and Hallerman 2009).

This huge volume of targeted ads is channeled to viewers via ad networks – intermediaries which package and sell ad space from multiple publishers' websites, video games, or other media vehicles. To match ads to appropriate audience segments, meet advertisers' goals, and ensure opportunities to serve advertising are not wasted, ad networks must solve complex planning problems. This paper focuses on the planning problem of an ad network which manages what we call *Guaranteed Targeted Display Advertising* (GTDA) – targeted advertising with the following properties:

- CPM Sales Model: Advertisers pay for a number of "eyeballs," called *impressions*. Each impression corresponds to an individual that sees an ad at a particular point in time, e.g., by viewing a banner ad on a webpage. Prices are quoted in cost-per-thousand (CPM); e.g., a $30 CPM means $30 buys 1000 impressions. This is a widely used sales model, often used to price webpage banner ads and dynamic ads in video games (cf. Surmanek 1995).
- Measurable Progress: The exact number of impressions served to date is known.
- Targeting Control: Ads shown to a specific individual can be chosen based on that individual's characteristics (demographic, geographic, and/or behavioral).
- Guaranteed Delivery: The network provider promises to serve each advertiser an agreed-upon number of impressions over a fixed time period. In this sense, delivery is "guaranteed," since ad networks do whatever they can to avoid under-delivery. Due to considerable uncertainty in the audience sizes of the various audience segments, ad networks must choose carefully when deciding which individuals get served which ads.

Although we assume all of these properties hold, model extensions can handle a broader class of targeted advertising, such as ads priced per click instead of per impression.

We solve the ad network's single-period planning problem that allocates impressions generated by multiple audience segments to multiple ad campaigns. The ad network prefers plans that yield both *high reach* and *low variance*. High reach means a large number of *unique* individuals see each ad campaign. Low variance means the outcome from executing the plan is highly predictable.

Specifically, the plan can be thought of as a policy the ad network follows when serving ads over the planning horizon; thus, due to audience uncertainty the number of impressions that get served to each campaign under this policy is a random variable, and the ad network prefers when the variance of the number of impressions served to each campaign is low.

We formulate the single-period planning problem as a transportation problem with a quadratic objective, and show that this formulation is good at spreading impressions proportionally across audience segments. Moreover, by explicitly modeling audience uncertainty, forecast errors, and the ad server's execution of the plan, we derive sufficient conditions for when proportionally spreading impressions minimizes variance and maximizes expected reach. Ad managers that proportionally spread impressions, as is common in practice, can use these conditions to verify the optimality of existing systems or, if some conditions are not met, identify opportunities for improvement.

From a computational standpoint, ad networks must solve planning problems with millions or even billions of audience segments. This arises because the subsets of viewers that different advertisers target can intersect in complex ways. Therefore, even when each advertiser targets only a handful of customer segments, the ad network must plan at a much finer resolution to account for the interactions among the full set of advertisers that have purchased intersecting blocks of audience. For example, if advertisers can choose to target 2 genders, 10 age categories, and 5 income levels in 500 geographic regions over 7 days and 6 dayparts in any conceivable way, the planning problem has $2 \times 10 \times 5 \times 500 \times 7 \times 6 = 2.1$ million viewer types. Yet over a short time window, impressions only come from a sparse subset of these viewer types. To address this issue, we develop two efficient algorithms for intelligently aggregating the audience space: the first assumes management specifies a fixed clustering of audience segments to use in an aggregate problem, and appropriately scales the aggregate solution to produce a feasible solution in the disaggregate space; the second intelligently refines the audience space into successively smaller clusters, converging to an optimal solution for the original disaggregate problem without ever explicitly considering the disaggregate problem. In many cases, near-optimal schedules can often be produced despite significant aggregation.

To summarize, this paper contributes to the ad planning literature by defining Guaranteed Targeted Display Advertising (GTDA) as a specific class of targeted advertising which cross-cuts various types of media and is characterized by four structural properties that allow ads to be planned in a common manner. We show that for GTDA, minimizing a quadratic objective is a good surrogate for maximizing reach and minimizing variance, and derive sufficient conditions for when reach and variance are exactly optimized. Furthermore, we introduce two algorithms to intelligently aggregate the potentially large audience space; these algorithms exploit the spreading which our quadratic objective induces.

Section 2 reviews the literature, §3 introduces the model, §4 formalizes reach and variance, and §5 derives sufficient conditions for when our quadratic objective is a good surrogate for reach and variance. Finally, §6 develops the aggregation theory for transportation problems with our quadratic objective, introduces two aggregation algorithms, and evaluates their performance.

2. Literature Review

Our problem – the single-period planning problem of allocating impressions from audience segments to ad campaigns – has been studied by, among others, Langheinrich et al. (1999) and Nakamura and Abe (2005). These authors also use a transportation problem formulation, but employ a linear objective rather than our quadratic objective. Their linear objective is appropriate for maximizing click-throughs in the absence of audience uncertainty; in our case, however, a linear objective would produce solutions that are not robust to audience uncertainty. Tomlin (2000) proposes maximizing entropy to find robust advertising allocations. As is known in the traffic modeling literature, maximizing entropy produces well-spread allocations that tend to be robust. We also advocate spreading impressions across audience segments; however, our quadratic objective, which is not equivalent to maxing entropy, has explicit connections to reach and variance in the context of GTDA, which we derive.

More recently, a quadratic objective like ours has been suggested by several researchers, notably Ghosh et al. (2009) and Yang et al. (2010), but for a different reason. These authors, as well as Tomlin et al. (2008), Yang and Tomlin (2008), and Tomlin et al. (2009) who presented earlier

work at professional conferences, focus on maximizing a quality they call "representativeness." The idea is that advertisers want impressions from *all* audience segments which match the targeting requirements specified, not just the particular subset of audience that is easiest (or cheapest) for the ad network to deliver. In this context, minimizing a quadratic objective spreads impressions across audience segments by minimizing the L2 distance to the "most representative" allocation; i.e. one that gives each campaign an equal proportion of each audience segment. Since we use the same objective, our formulation also maximizes representativeness; however, we focus on determining when reach and variance are maximized as a result of maximizing representativeness.

The algorithms we introduce in §6 to aggregate the audience space of large GTDA problems contribute to the aggregation theory of math programs. In the context of transportation planning, Zipkin (1980a) defines the basic aggregation framework, which is refined by Zipkin and Raimer (1983). In our case, a crucial assumption of Zipkin (1980a) doesn't hold: The adjacency structure of all aggregated nodes needs to be the same to use Zipkin's algorithm, whereas in GTDA problems *all* nodes have a unique adjacency structure. Because this assumption is violated, solutions of the aggregate problem may be infeasible after disaggregation. Much of what we do focuses on restoring feasibility; thus, our work contributes to the aggregation literature for the special case of quadratic transportation problems that arise in GTDA planning. We should note that in addition to Zipkin (1980a), aggregation has been studied in the context of generalized transportation networks (Litvinchev and Rangel 2006), linear programs (Zipkin 1980b, Zipkin 1980c, and Leisten 1997), stochastic programs (Birge 1985, Wright 1994), and convex network programs (Zipkin 1982). The surveys by Rogers et al. (1991) and Vakhutinsky et al. (1979) list the bounds known at that time, and the book by Litvinchev and Tsurkov (2003) is also a good reference.

The paper by Walsh et al. (2010) deserves mention since the authors also study how to intelligently partition the potentially large audience space in the context of a targeted advertising problem. The planning problem that they study, however, is substantially different from the GTDA planning problem. Specifically, Walsh et al. (2010) maximize the revenue generated from serving ads over time, given budget-constrained bidders which can have different bids for different audience

segments. The authors maximize a linear revenue objective, which from our experience results in allocations that are poorly spread across audience segments, and therefore have low reach, high variance, and low representativeness. While Walsh et al. (2010) develop a column generation scheme for their particular ad planning problem, our aggregation algorithms exploit the quadratic objective and impression goal constraints which are crucial components of GTDA planning problems.

Finally, there is a significant body of online advertising literature which is tangential to this paper. Problems studied include the design of incentive-compatible auctions for allocating advertising (Edelman et al. 2007), allocating advertising to maximize revenue subject to budget constraints using online algorithms (Mehta et al. 2007), packing 2D ads of different shapes and sizes into 2D areas (Adler et al. 2002, Dawande et al. 2003, Menon and Amiri 2004, Kumar et al. 2006), revenue management (Roels and Fridgeirsdottir 2009), and pricing (Araman and Fridgeirsdottir 2008, Fridgeirsdottir and Najafi-Asadolahi 2008). Applications include scheduling ads in video games (Turner et al. 2011) and in cellphone text messages (De Reyck and Degraeve 2003). As well, a substantial body of literature exists in marketing journals, where the focus has been on solving the single advertiser's problem of allocating impressions across media vehicles rather than the ad network's problem of allocating multiple advertisers' impressions across audience segments; for a good review, see Danaher (2008).

3. The Model

3.1. Definitions and Notation

We study the single-period planning problem of a *network provider* serving ads to multiple advertisers. A content publisher (e.g., a website such as `espn.com`) can be a network provider, but oftentimes ad networks such as DoubleClick fill this role by aggregating ad space from multiple publishers. The network provider manages *ad inventory*, which refers to the *impressions*, i.e. eyeballs or page views, generated by the content's audience. Advertisers buy impressions from the network provider by purchasing a *campaign*: a contract that specifies the number of impressions to be served over a fixed time period. A campaign's *targeting* constrains the inventory it can be allocated to specific audience segments, which we call *viewer types*. For example, a campaign may want only viewers

from specific media assets (e.g., `www.espn.com/golf/`), time periods (e.g., Saturday evening), geographic regions (e.g., France), demographic profiles (e.g., male, age 18-25), or behavioral categories (e.g., golfers). The number of impressions a viewer type generates is called its *supply*. We use the following notation:

- V = the set of all viewer types
- K = the set of all campaigns
- V_k = the set of viewer types that campaign k targets
- K_v = the set of campaigns that target viewer type v
- g_k = the impression goal of campaign k
- S_v = the supply of viewer type v (a random variable)
- $s_v := \mathbb{E}[S_v]$ = the expected supply of viewer type v
- $c_v := 1/s_v$ = the reciprocal of the expected supply of viewer type v

The *viewer type partition* is the partition of the audience space induced by the targeting constraints of all campaigns managed by the network provider.

EXAMPLE 1. *Figure 3a displays a viewer type partition induced by three campaigns: campaign A targets Pittsburgh (viewer types a, d, e, and g), campaign B targets males (viewer types b, d, f, and g), and campaign C targets the 18-25 year old age group (viewer types c, e, f, and g). The viewer types in this example are: a={from Pittsburgh, female, not aged 18-25}, b={not from Pittsburgh, male, not aged 18-25}, c={not from Pittsburgh, female, aged 18-25}, d={from Pittsburgh, male, not aged 18-25}, e={from Pittsburgh, female, aged 18-25}, f={not from Pittsburgh, male, aged 18-25}, and g={from Pittsburgh, male, aged 18-25}.*

We cast the planning problem as a quadratic transportation problem: we represent each viewer type as a source node, each campaign as a sink node, and for each viewer type that a campaign targets, we connect the corresponding source to sink with an uncapacitated arc. There are two equivalent formulations of this planning problem (PP): an *impression formulation* (PP^{IMP}) in which the amount of flow on arc (v, k) represents the absolute number of impressions of viewer

type v allocated to campaign k, and a *proportion formulation* (PP^{PROP}) in which the amount of flow on arc (v,k) represents the proportion of viewer type v allocated to campaign k:

$$
\begin{array}{rlll}
(PP^{IMP}) \min & \sum_{k\in K, v\in V_k} c_v x_{vk}^2 & & \\
\text{s.t.} & \sum_{v\in V_k} x_{vk} & = g_k \ \forall k \in K & \text{(impression goals)} \\
& \sum_{k\in K_v} x_{vk} & \leq s_v \ \forall v \in V & \text{(supply constraints)} \\
& x_{vk} & \geq 0 \quad \forall k \in K, v \in V_k & \text{(nonnegativity)} \\
(PP^{PROP}) \min & \sum_{k\in K, v\in V_k} s_v p_{vk}^2 & & \\
\text{s.t.} & \sum_{v\in V_k} s_v p_{vk} & = g_k \ \forall k \in K & \text{(impression goals)} \\
& \sum_{k\in K_v} p_{vk} & \leq 1 \quad \forall v \in V & \text{(supply constraints)} \\
& p_{vk} & \geq 0 \quad \forall\, k \in K, v \in V_k & \text{(nonnegativity)}
\end{array}
$$

The impression formulation (PP^{IMP}) has a deterministic interpretation: *Assuming* supply is known (i.e. $S_v = s_v$), the decision variable x_{vk} measures the number of impressions of viewer type v allocated to campaign k. In this case, the impression goal constraint ensures each campaign is allocated exactly the number of impressions it requires, and the supply constraint ensures none of the viewer types are overallocated. The quadratic objective tends to spread impressions proportionally across all viewer types that a campaign targets; we will see in §5 why this is important.

The proportion formulation (PP^{PROP}) uses scaled decision variables $p_{vk} = x_{vk}/s_v$; the advantage being that for every *realization* of S_v, we can consider $S_v p_{vk}$ to be the number of impressions generated by viewer type v that get served to campaign k. Note that $\mathbb{E}[S_v p_{vk}] = s_v p_{vk} = x_{vk}$; i.e. x_{vk} is the number of impressions of viewer type v that will be served to campaign k *in expectation.* Indeed, when S_v is uncertain, x_{vk} should be interpreted as an allocation achieved in expectation. In this sense, (PP) is similar to a portfolio optimization problem in which asset weights are selected to minimize variance subject to meeting a desired expected return. In our case, the impression goal constraint is analogous to the portfolio optimization's expected return constraint, and states that in expectation, each campaign must be served exactly the number of impressions it requires. Although we analyze (PP) as if it is a single-period problem, it is important to note that re-solving (PP) with sufficient frequency is a good way to serve impressions uniformly over time.

3.2. Model of Audience Uncertainty

The network provider's *ad server* is the computer system responsible for selecting which ads to serve at each point in time. We say an *arrival* occurs at the instant the ad server must select and push ads to a single viewer. In the context of webpage banner ads, an arrival occurs when a viewer loads a webpage in their browser; at this point, the ad server selects one ad for each of the n banner ad slots on the page. In the context of video games, an arrival occurs when a player loads a new game level; at this point, the ad server selects one ad for each of the n ad slots in the level.

We assume viewers of type v arrive into the system according to a Poisson process with rate λ_v. This is reasonable, since a nonhomogeneous Poisson process can accurately model arrivals to home pages of websites (Liu et al. 2001, Chlebus and Brazier 2007) as well as arrivals of people playing video games (Turner 2010). Furthermore, we assume each arrival r of viewer type v generates an i.i.d. random number of impressions Y_v^r with mean μ_v and standard deviation σ_v. Therefore, the supply of viewer type v can be written as:

$$S_v = Y_v^1 + \cdots + Y_v^{M_v}, \text{ where } M_v \sim \text{ Poisson}(\lambda_v).$$

For banner ads, often exactly one impression is counted for each ad served, thus $Y_v^r = n$ when there are n ad slots on a webpage. But more generally, impressions can be counted in different ways, e.g., logging an impression once an ad is on-screen for 10 seconds, as is the case in some video games. Since S_v is a compound Poisson[1] random variable, its expectation and variance are:

$$\mathbb{E}[S_v] = \mu_v \lambda_v, \ \text{Var}[S_v] = \left(\mu_v^2 + \sigma_v^2\right) \lambda_v. \tag{1}$$

3.3. Model of Ad Server Execution

Consider what happens when the ad server processes a single arrival of viewer type v. To exactly track the plan (PP), the ad server would like to assign a p_{vk}-fraction of the impressions that this arrival will generate to campaign k. However, such an exact assignment is usually not possible,

[1] A random variable Y is compound Poisson if it can be written as the sum $Y = \sum_{n=1..N} X_n$, where N is a Poisson random variable and the X_n's are i.i.d. Since N and X_n are independent, from first principles we have $\mathbb{E}[Y] = \mathbb{E}[N]\mathbb{E}[X]$ and $\text{Var}[Y] = \mathbb{E}[N](\text{Var}[X] + \mathbb{E}[X]^2)$.

because each arrival has a discrete number of slots, and, in general, each slot generates a random number of impressions. In particular, if the number of ad slots is less than the number of campaigns k with $p_{vk} > 0$, the ad server must randomly pick a subset of the eligible campaigns to serve to this arrival. Modeling these details allows us to measure how well the ad server executes the plan.

Let F^r_{vk} be the *random fraction* of the impressions generated by the r^{th} arrival of viewer type v that get served to campaign k. Since the ad server is trying to execute the plan as closely as possible, we require $\mathbb{E}[F^r_{vk}] = p_{vk}$ to hold for any campaign k that, according to the plan, is scheduled to receive a p_{vk}-fraction of viewer type v's impressions.

If any two campaigns, according to the plan, should be served the same p-fraction of viewer type v's impressions, we require the ad server to treat these two campaigns in a symmetric fashion. Therefore, for all campaigns which the plan allocates a p-fraction of viewer type v, we assume the random variables F^r_{vk} are identically distributed over all arrivals r and campaigns k. This allows us to define $\alpha_v(p) := \mathbb{E}[(F^r_{vk})^2]$ as the second moment of the random fraction of impressions awarded to a campaign that is allotted a p-fraction of viewer type v. As well, we define $\beta_{vk} := \text{Prob}(F^r_{vk} > 0)$ as the probability that campaign k is served in one or more ad slots of an arrival of type v. Since we require campaigns that are allocated the same p-fraction of viewer type v to be treated the same way, we often write $\beta_v(p)$ in place of β_{vk}.

EXAMPLE 2. *Banner ads are being served by a web server. Arrivals of all viewer types see the same webpage, which has n ad slots to be filled, and we assume that each of the n slots will generate exactly one impression. For an arrival of type v, the ad server would like to assign campaign k to np_{vk} ad slots; however, np_{vk} may not be integral, and so the number of slots assigned to campaign k is rounded up or down. The random fractions are defined as:*

$$F^r_{vk} = \begin{cases} \frac{\lfloor np_{vk} \rfloor}{n} & \text{with probability } \lfloor np_{vk} \rfloor + 1 - np_{vk} \\ \frac{\lceil np_{vk} \rceil}{n} & \text{with probability } np_{vk} - \lfloor np_{vk} \rfloor \end{cases} \tag{2}$$

It is straightforward (see §EC.1) to compute $\mathbb{E}[F^r_{vk}] = p_{vk}$; $\alpha_v(p) = \frac{2\lfloor np \rfloor + 1}{n}(p - \frac{\lfloor np \rfloor}{n}) + \frac{\lfloor np \rfloor^2}{n^2}$; and $\beta_v(p) = \min(np, 1)$. Note that $\alpha_v(p)$ is piecewise-linear convex increasing with n segments, while $\beta_v(p)$ is piecewise-linear concave increasing with 2 segments.

In Example 2, we allow the possibility of serving the same ad in more than one slot. The choice to permit such "multi-impressions" as Nakamura and Abe (2005) call them depends on the desired look of the website or medium. For example, in video games, it is usually good to serve the same ad in multiple slots, since not all slots end up being seen during game play. Note that if F^r_{vk} as defined in (2) is used to model how the ad server does its low-level ad slotting, then multi-impressions can be avoided by adding the constraint $p_{vk} \leq 1/n$ to (PP^{PROP}), since $p_{vk} \leq 1/n \implies F^r_{vk} \leq 1/n$.

Our framework assumes the plan provides high-level guidance to the ad server, but does not fully specify how the ad server should execute the plan. The alternative is to generate plans that also provide lower-level direction to the ad server; i.e. plans that implicitly or explicitly include the exact allocation of ads to slots for each arrival. Abrams et al. (2008) call the pattern of ads an individual viewer is served a *slate*, and generate a plan with one or more slates for each viewer type. When their plan is executed, slate i is shown with probability f_i, exposing the viewer to the set of campaigns in the slate. In our framework, F^r_{vk} can be defined to model the random selection of slates; however, since we require all campaigns assigned the same p-fraction of a viewer type to be treated in a symmetric fashion, the slates must be generated in a way that ensures this symmetry.

In practice, it may be difficult to choose a suitable analytical expression for F^r_{vk} that closely matches the complex ad slotting heuristics of a given ad server; however, historical data can be used to estimate the shape of the $\alpha_v(p)$ and $\beta_v(p)$ functions, which are the crucial model components.

4. Performance Metrics: Reach and Variance

We now derive expressions for reach and variance. Later in §5, we will show that solutions to (PP) tend to have high reach and low variance.

4.1. Reach

Let U_{vk} be the *reach* of campaign k in viewer type v – i.e. the number of unique individuals from viewer type v that see campaign k over a fixed time period. To derive an expression for *expected reach* $\mathbb{E}[U_{vk}]$, we assume the population of viewers (across all viewer types) is homogeneous, so that each individual viewer arrives at rate η (this permits a concrete example, however our results hold

for heterogeneous populations so long as the reach function (4) remains concave increasing in $\beta_v(p)$ and is directly proportional to λ_v). Let m_v be the number of individuals that belong to viewer type v, and let $A_v^i \sim \text{Poisson}(\eta)$ be the number of arrivals of individual i in viewer type v. Then by definition, $\lambda_v = \mathbb{E}[A_v^1 + \cdots + A_v^{m_v}] = m_v \mathbb{E}[A_v^i] = m_v \eta \implies m_v = \lambda_v/\eta$. Furthermore, for each arrival of individual i, campaign k is served with probability β_{vk}; thus, individual i sees campaign k in $W_{vk}^i \sim \text{Poisson}(\eta\beta_{vk})$ arrivals. Using the indicator variable $Z_{vk}^i := \{1 \textit{ if } W_{vk}^i \geq 1,\ 0 \textit{ otherwise}\}$, we have $U_{vk} := \sum_{i=1..m_v} Z_{vk}^i \sim \text{Binomial}(m_v, \text{Prob}(Z_{vk}^i = 1)) \equiv \text{Binomial}(\lambda_v/\eta, 1 - e^{-\eta\beta_{vk}})$. Therefore,

$$\mathbb{E}[U_{vk}] = \frac{\lambda_v}{\eta}\left(1 - e^{-\eta\beta_{vk}}\right), \tag{3}$$

so the expected reach of a campaign allocated a p-fraction of viewer type v is

$$\mathbb{E}[U_v(p)] = \frac{\lambda_v}{\eta}\left(1 - e^{-\eta\beta_v(p)}\right). \tag{4}$$

Finally, we assume the reach of campaign k, denoted U_k, is simply $U_k \stackrel{\text{def}}{=} \sum_{v \in V_k} U_{vk}$; i.e. reach is additive across viewer types. Although this assumption is common in the media industry (see, for example, Surmanek 1995), in general $U_k \leq \sum_{v \in V_k} U_{vk}$ due to *audience duplication*: individual viewers may belong to more than one viewer type, and thus end up double-counted; e.g., if viewer type 1 is {Males that visit `www.espn.com/golf/`} and viewer type 2 is {Males that visit `http://finance.yahoo.com/`}, some viewers fall into both categories, yet should only be counted once. Of course, when viewer types are defined using only properties of audience members, and not properties of media vehicles, audience duplication does not occur and our definition of U_k is exact.

The *total expected reach* of plan (PP) with solution $\mathbf{p} = \{p_{vk} : k \in K, v \in V_k\}$ is the expected reach summed up across all campaigns:

$$\mathbb{E}[U(\mathbf{p})] = \sum_{k \in K} \mathbb{E}[U_k] = \sum_{k \in K, v \in V_k} \mathbb{E}[U_v(p_{vk})] = \sum_{k \in K, v \in V_k} \frac{\lambda_v}{\eta}\left(1 - e^{-\eta\beta_v(p_{vk})}\right). \tag{5}$$

4.2. Variance

Let X_{vk} be the actual number of impressions served to campaign k from viewer type v. If λ_v is known, then X_{vk} is a random variable of the form:

$$X_{vk} = \sum_{r=1..M_v} F_{vk}^r Y_v^r, \text{ where } M_v \sim \text{ Poisson}(\lambda_v). \tag{6}$$

To compute $\mathbb{E}[X_{vk}]$ and $\mathrm{Var}[X_{vk}]$, we need estimates for the first and second moments of F^r_{vk} and Y^r_v, as well as an estimate for λ_v. We assume the only parameter subject to significant estimation error is λ_v; i.e. we have a reasonable understanding of the dynamics of the ad server on which the first and second moments of F^r_{vk} and Y^r_v depend, but audience size is subject to forecast error. To model the estimation error of λ_v, we assume a forecasting system uses historical data and forward-looking statements from management to compute the *estimator* Λ_v. We insist that Λ_v is unbiased ($\mathbb{E}[\Lambda_v] = \lambda_v$), and that $\mathrm{Var}[\Lambda_v]$ – the variance of the forecasting system's point estimate for λ_v – can be computed. With forecast error of audience size accounted for, Equation (6) generalizes to:

$$X_{vk} = \sum_{r=1..M_v} F^r_{vk} Y^r_v, \text{ where } M_v \sim \text{ Poisson}(\Lambda_v). \tag{7}$$

Using this definition of X_{vk}, the mean and variance of the number of impressions served to campaign k from viewer type v are (see §EC.2 for the derivations):

$$\mathbb{E}[X_{vk}] = \mu_v \lambda_v p_{vk} = s_v p_{vk} \tag{8}$$

$$\mathrm{Var}[X_{vk}] = \underbrace{(\sigma_v^2 + \mu_v^2)\lambda_v \alpha_v(p_{vk})}_{\text{Stochastic Variance}} + \underbrace{\mu_v^2 p_{vk}^2 \mathrm{Var}[\Lambda_v]}_{\text{Forecast Variance}}. \tag{9}$$

In general, $\mathrm{Var}[X_{vk}]$ has two components: *forecast variance* caused by uncertainty in the forecasted arrival rate; and *stochastic variance* which, assuming a known arrival rate, is caused by uncertainty in the number of arrivals, the number of impressions per arrival, and the number of impressions assigned to each campaign from each arrival.

Example 3. *A forecasting system uses τ periods of historical data from a server log to compute Λ_v; i.e. audience size distributions are assumed stationary. Since server logs can be very large, we use a sampled log where each arrival in the full log is sampled independently with probability γ. Letting Z^t_v be the number of arrivals of viewer type v from period t in the sampled log, we treat Z^t_v as an estimator – a random variable that encapsulates the variation in the number of arrivals that could have occurred. In this case, Λ_v – the maximum likelihood estimator for the arrival rate parameter λ_v under the assumption that arrivals are i.i.d. Poisson(λ_v) in all periods – is computed by averaging the number of sampled arrivals Z^t_v over time periods $t = 1..\tau$ and scaling by γ:*

$$\Lambda_v = \frac{1}{\gamma\tau} \sum_{t=1..\tau} Z^t_v, \text{ where } Z^t_v \sim \text{ Poisson}(\gamma\lambda_v), \tag{10}$$

$$\mathbb{E}[\Lambda_v] = \frac{1}{\gamma\tau}\sum_{t=1..\tau}\mathbb{E}[Z_v^t] = \frac{1}{\gamma\tau}\times\tau(\gamma\lambda_v) = \lambda_v, \text{ and} \tag{11}$$

$$\mathrm{Var}[\Lambda_v] = \frac{1}{\gamma^2\tau^2}\sum_{t=1..\tau}\mathrm{Var}[Z_v^t] = \frac{1}{\gamma^2\tau^2}\times\tau(\gamma\lambda_v) = \frac{\lambda_v}{\gamma\tau}; \tag{12}$$

and the variance of the number of impressions served to campaign k from viewer type v is computed by substituting (12) into (9):

$$\mathrm{Var}[X_{vk}] = \underbrace{(\sigma_v^2+\mu_v^2)\lambda_v\alpha_v(p_{vk})}_{\text{Stochastic Variance}} + \underbrace{\frac{\mu_v s_v p_{vk}^2}{\gamma\tau}}_{\text{Forecast Variance}}. \tag{13}$$

Finally, we define $X_k = \sum_{v\in V_k} X_{vk}$ as the total number of impressions served to campaign k, and $X(\mathbf{p}) = \sum_{k\in K} X_k$ as the total number of impressions served to all campaigns under the plan (PP). Since variance is additive, the *total stochastic variance* of plan (PP) is:

$$\mathrm{StochVar}[X(\mathbf{p})] = \sum_{k\in K}\mathrm{StochVar}[X_k] = \sum_{k\in K, v\in V_k}\mathrm{StochVar}[X_{vk}] = \sum_{k\in K, v\in V_k}(\sigma_v^2+\mu_v^2)\lambda_v\alpha_v(p_{vk}), \tag{14}$$

and the *total forecast variance* of plan (PP) is:

$$\mathrm{ForecastVar}[X(\mathbf{p})] = \sum_{k\in K}\mathrm{ForecastVar}[X_k] = \sum_{k\in K, v\in V_k}\mathrm{ForecastVar}[X_{vk}] = \sum_{k\in K, v\in V_k}\mu_v^2 p_{vk}^2\mathrm{Var}[\Lambda_v]. \tag{15}$$

5. Model-Based Results

This section derives sufficient conditions for the solution of (PP) to a) maximize expected reach, b) minimize stochastic variance, and c) minimize forecast variance. We begin by plotting the main functions of interest – $\alpha(p)$, $\beta(p)$, expected reach, stochastic variance, and forecast variance – to provide intuition for the results that follow. The sufficient conditions are then derived, and followed with a discussion of when they can be expected to hold in practice.

5.1. Visualizing The Important Functions

Consider an example in which ad server execution is modeled by (2), Λ_v is defined by (10), and parameters are fixed at $\gamma = 0.1$, $\tau = 3$, $\mu = 5$, $\sigma = 5$, $\eta = 2.2$, $\lambda_v = 10$, and $n = 3$. Figure 1 plots several important functions of the proportion $p \equiv p_{vk}$ of viewer type v allocated to a single campaign k. Since there are $n = 3$ ad slots, $\alpha(p)$ and $\mathrm{StochVar}(p)$ have $n = 3$ piecewise-linear segments. The

kinks in $\alpha(p)$ coincide with the proportions $p \in \{\frac{0}{n}, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n}\} = \{\frac{0}{3}, \frac{1}{3}, \frac{2}{3}, \frac{3}{3}\}$ where the ad server can assign the campaign to exactly np slots in each arrival. Between each pair of kinks, $\alpha(p)$ is linear since the ad server randomizes between giving this campaign $\lfloor np \rfloor$ and $\lceil np \rceil$ slots, and the probability of rounding the number of slots up grows linearly with p. Finally, StochVar(p) is piecewise-linear since it is a scaled version of $\alpha(p)$.

We also see that $\beta(p)$ and Reach(p) increase until $p = 1/n = 1/3$, beyond which expected reach cannot increase because $\beta(p) = 1$ implies all individuals of this viewer type see this campaign. As well, we see ForecastVar(p) is quadratic in p, and is independent of $\alpha(p)$ and $\beta(p)$.

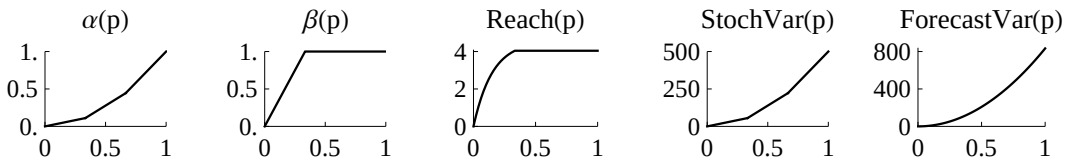


Figure 1 Important functions of the proportion $p \equiv p_{vk}$ of viewer type v allocated to campaign k.

5.2. Sufficient Conditions for the Quadratic Objective to Optimize Reach and Variance

Let $S_k := \sum_{v \in V_k} S_v$ be the total supply (in impressions) that campaign k targets, and let $q_k := g_k / \mathbb{E}[S_k]$. We define the *equal-proportion allocation* $\mathbf{q}$ as the solution to (PP^{PROP}) which spreads impressions proportionally across all targeted viewer types; i.e. $\{p_{vk} = q_k \; \forall k \in K, v \in V_k\}$. The following theorem proves optimality of the equal-proportion allocation in a broad problem class.

THEOREM 1. *Consider the objective function $f(\mathbf{p}) = \sum_{k \in K, v \in V_k} s_v h(p_{vk})$, where $h : \mathbb{R} \to \mathbb{R}$ is convex (but possibly nondifferentiable). The equal-proportion allocation $\mathbf{p} = \mathbf{q}$ is optimal for the problem:*

$$
\begin{aligned}
(P1) \; \min \; & f(\mathbf{p}) \\
s.t. \; & \textstyle\sum_{v \in V_k} s_v p_{vk} = g_k \;\; \forall k \in K && \textit{(impression goals)} \\
& p_{vk} \geq 0 \qquad\qquad \forall k \in K, \forall v \in V_k && \textit{(non-negativity)}
\end{aligned}
$$

Proof. Decomposing ($P1$) by campaign, Jensen's Inequality implies the equal-proportion allocation is optimal for each campaign's subproblem. For details, see §EC.3 □

Problem (*P1*) is closely related to (PP^{PROP}): If (PP^{PROP}) is uncapacitated, then it is an instance of (*P1*) with $h(p) \stackrel{\text{def}}{=} p^2$. Therefore, if the equal-proportion allocation is feasible in (PP^{PROP}), supply constraints are nonbinding and by Theorem 1 the equal-proportion allocation is optimal. On the other hand, when supply constraints bind and the equal-proportion allocation is infeasible, the optimal solution of (PP^{PROP}) is *close* to the equal-proportion allocation. This is because, as the following proposition shows, the quadratic objective of (PP^{PROP}) minimizes what can be viewed as a L2 distance to the equal-proportion allocation.

PROPOSITION 1. *Let* $d(\mathbf{a},\mathbf{b}) = \sqrt{\sum_{k\in K, v\in V_k} s_v(a_{vk}-b_{vk})^2}$ *be a simple modification to the usual L2 distance that accounts for the fact that viewer types with a larger audience size should be given more weight*[2]*. Then the objectives* $\min \sum_{k\in K, v\in V_k} s_v p_{vk}^2$ *and* $\min d(\mathbf{p},\mathbf{q})$ *are equivalent for* (PP^{PROP}).

Proof. Since d is nonnegative, $\min d(\mathbf{p},\mathbf{q})$ and $\min d(\mathbf{p},\mathbf{q})^2$ are equivalent. Moreover,

$$\begin{aligned} d(\mathbf{p},\mathbf{q})^2 &= \sum_{k\in K, v\in V_k} s_v(p_{vk}-q_{vk})^2 = \sum_{k\in K, v\in V_k} s_v(p_{vk}^2 - 2q_k p_{vk} - q_k^2) \\ &= \sum_{k\in K, v\in V_k} s_v p_{vk}^2 - 2\sum_{k\in K} q_k \sum_{v\in V_k} s_v p_{vk} - \sum_{k\in K, v\in V_k} s_v q_k^2 \\ &= \sum_{k\in K, v\in V_k} s_v p_{vk}^2 - 2\sum_{k\in K} q_k g_k - \sum_{k\in K, v\in V_k} s_v q_k^2, \end{aligned}$$

where the last line follows by substituting the impression goals $g_k = \sum_{v\in V_k} s_v p_{vk}$. Minimizing the last line is equivalent to minimizing $\sum_{k\in K, v\in V_k} s_v p_{vk}^2$ since the last two terms are constants. □

Under conditions which we will describe, the expressions for expected reach, stochastic variance, and forecast variance match the functional form of the objective in (*P1*). Therefore, when these conditions hold, the equal-proportion allocation is optimal with respect to reach and variance for uncapacitated cases of (PP^{PROP}). Moreover, in the capacitated case the optimal solution to (PP^{PROP}) is *close* to the equal-proportion allocation, motivating the use of the quadratic objective as a surrogate for reach and variance. Figure 2 summarizes the sufficient conditions which imply the optimal solution to (PP) maximizes reach, minimizes stochastic variance, and minimizes forecast

[2] If the summation present in $d(\mathbf{a},\mathbf{b})$ is viewed as a summation over *impressions* instead of viewer types, then this is the classic L2 distance. Formally, the inner product on which this distance function is based is $\langle \mathbf{a},\mathbf{b}\rangle_S = \sum_{k\in K, v\in V_k} s_v a_{vk} b_{vk}$; thus, the corresponding norm is $||\cdot||_S = \sqrt{\langle\cdot,\cdot\rangle_S}$, and we have $d(\mathbf{a},\mathbf{b}) = ||\mathbf{a}-\mathbf{b}||_S$.

variance. For derivations of these conditions, see §EC.3.1. Note that forecast variance is optimal even when capacity constraints bind, since under the conditions listed in Figure 2, the expression for forecast variance simplifies to the exact quadratic objective used in (PP).

	Expected Reach Maximized	StochVar Minimized	ForecastVar Minimized
1. Homogeneous mean imps/arrival ($\mu_v = \mu \, \forall v \in V$)	✔	✔	✔
2. Homogeneous stdev imps/arrival ($\sigma_v = \sigma \, \forall v \in V$)		✔	
3. Homogeneous convex $\alpha_v(p)$		✔	
4. Homogeneous concave $\beta_v(p)$	✔		
5. $\text{Var}[\Lambda_v] = \theta \mathbb{E}[\Lambda_v] \, \forall v \in V$ for some constant $\theta \geq 0$			✔
6. Uncapacitated (nonbinding supply constraints)	✔	✔	

Figure 2 Read down a column to find the conditions which imply (PP) optimizes that column's objective. For example, if all viewer types share the same mean number of impressions per arrival ($\mu_v = \mu \, \forall v \in V$), all viewer types share the same $\beta(p)$ function ($\beta_v(p) = \beta(p) \, \forall v \in V$), the function $\beta(p)$ is concave, and supply constraints are nonbinding, then any optimal solution to (PP) maximizes expected reach.

Many of the conditions listed in Figure 2 hold in practice. For example, if the ad server's execution is modeled by Equation (2), then $\alpha(p)$ is convex and $\beta(p)$ is concave, as seen in panels 1 and 2 of Figure 1. Furthermore, when arrival rates are stationary, i.e. if Λ_v is computed from τ periods of historical data sampled with frequency γ as in Example 3, then the assumption "$\text{Var}[\Lambda_v] = \theta\lambda_v \, \forall v \in V$ for some constant $\theta \geq 0$" holds with $\theta = 1/\gamma\tau$.

Conditions that require viewer types to be (partially) homogeneous can also be accommodated in practice. This is because the network provider's full planning problem is realistically multi-period and involves multiple objectives and idiosyncratic constraints. In this broader context, a math programming approach to planning necessarily involves problem decomposition. If the decomposition is designed such that each subproblem looks like (PP) and has homogeneous viewer types, then variance and reach can be optimized locally for each subproblem. Sets of viewer types likely to be homogeneous include all audience segments from websites with the same number of ad slots, and all audience segments from all video games that induce a similar pattern of game play.

6. Aggregation

In this section, we study aggregation of the viewer type space as a way to manage the potentially large number of viewer types, many of which correspond to very small populations of viewers. In the

context of transportation planning, Zipkin (1980a) introduces the aggregation framework, which is refined by Zipkin and Raimer (1983): An Aggregated Transportation Problem (ATP) is produced from an original Transportation Problem (TP) by grouping source and destination nodes into aggregated mega-source and mega-destination nodes, and adjusting the arc costs and capacities accordingly. An optimal solution for (ATP) is disaggregated into a "good" feasible solution for (TP), and the quality of this solution is assessed using a duality-based bound. The advantage of using an aggregation algorithm is that near-optimal solutions to (TP) can be found by solving (ATP), which is a much smaller problem.

In our case, a crucial assumption of Zipkin (1980a) does not hold: namely, the adjacency structure of all aggregated nodes should be the same. Without this assumption, solutions of (ATP) may be infeasible for (TP) after disaggregation. Much of this section focuses on restoring feasibility.

In terms of exposition, we deviate from Zipkin (1980a) by including the disaggregation formula which transforms an aggregate solution into a disaggregate solution *explicitly* in the aggregate problem. Therefore, the aggregate problem of Zipkin (1980a) is analogous to what we define as an *auxiliary transportation problem*, and the extended formulation that we call the aggregate problem is not explicitly defined by Zipkin (1980a). Furthermore, we only consider aggregation of viewer types, and not campaigns; this is because the viewer type space grows exponentially in the number of consumer characteristics, and is therefore the most important dimension for us to aggregate.

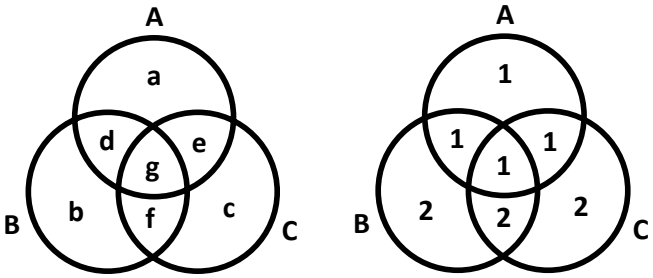
6.1. Notation and Definitions

Aggregation is accomplished by clustering viewer types into groups, which we call *inventory blocks*. An *inventory block partition* is a clustering in which each viewer type is assigned to exactly one inventory block. We extend the notation of §3.1 as follows:

- I = the set of all inventory blocks
- $i(v)$ = the inventory block to which viewer type v belongs
- V_i = the set of viewer types in inventory block i
- V_{ik} = the set of viewer types that campaign k targets in inventory block i
- I_k = the set of inventory blocks that campaign k targets

- K_i = the set of campaigns that target inventory block i

For example, Figure 3b shows one possible inventory block partition of the viewer type space introduced in Example 1; in this partition, all inventory from viewers in Pittsburgh is considered inventory block 1, while all remaining inventory is considered inventory block 2. Thus, $i(a) = i(d) = i(e) = i(g) = 1$, $i(b) = i(c) = i(f) = 2$, $V_2 = \{b, c, f\}$, $V_{2B} = \{b, f\}$, $K_2 = \{B, C\}$, and $I_B = I = \{1, 2\}$.



Viewer type partition Inventory block partition

Figure 3 Viewer type and inventory block partitions.

The supply of inventory block i is $S_i = \sum_{v \in V_i} S_v$; thus $s_i := \mathbb{E}[S_i] = \sum_{v \in V_i} \mathbb{E}[S_v] = \sum_{v \in V_i} s_v$. Analogously, the supply of the *subset* of inventory block i that campaign k targets is $S_{ik} = \sum_{v \in V_{ik}} S_v$; thus, the expected supply of inventory block i *available* to campaign k is $s_{ik} := \mathbb{E}[S_{ik}] = \sum_{v \in V_{ik}} s_v$.

Recall from §3.1 that the planning problem (PP) has two equivalent formulations: the impression formulation, which uses decision variables x_{vk}, and the proportion formulation, which uses decision variables $p_{vk} \stackrel{\text{def}}{=} x_{vk}/s_v$. In this section, we refer to (PP) as the *Original Planning Problem* (OPP), and write (OPP^{IMP}) and (OPP^{PROP}) for the impression and proportion formulations respectively.

One way of aggregating (OPP) is by adding constraints of the form $p_{vk} = p_{i(v),k}$; i.e. replacing each occurrence from the set of variables $\{p_{vk} : v \in V_{ik}\}$ with the new variable $p_{ik} \equiv p_{i(v),k}$, thereby forcing the proportional allocations for each viewer type within an inventory block to be the same. We take a slightly more general approach. Instead, we consider the *Aggregate Planning Problem* (APP) produced from (OPP) by adding the constraints $y_v = \min\left(1, 1/\sum_{k \in K_v} p_{i(v),k}\right)\ \forall v \in V$, and $p_{vk} = y_v p_{i(v),k}\ \forall k \in K, v \in V_k$. The quantity y_v is called the *yield* of viewer type v, and the *disaggregation formula* $p_{vk} = y_v p_{i(v),k}$ is used to convert from inventory block weights p_{ik} to viewer type proportions p_{vk} (we prefer to call p_{ik} a *weight* rather than a proportion, since $p_{ik} > 1$ is

allowed). Note that we do not intend to solve (APP) directly, but will use it to structurally compare different solution approaches.

EXAMPLE 4. *Consider an instance with 2 viewer types and 2 campaigns. Viewer types v and w each supply 1 impression in expectation ($s_v = s_w = 1$). Campaign A targets viewer type v only, while campaign B targets both viewer types v and w; thus, $V_A = \{v\}$, $V_B = \{v, w\}$, $K_v = \{A, B\}$, and $K_w = \{B\}$. Impression goals are $g_A = 3/4$ and $g_B = 1$. The solution $\{p_{vA} = 3/4,\ p_{vB} = 1/4,\ p_{wB} = 3/4\}$ is optimal for (OPP^{PROP}). This follows because, first, all feasible solutions have $p_{vA} = 3/4$, reducing the problem to $\{\min p_{vB}^2 + p_{wB}^2 \ \ s.t.\ p_{vB} + p_{wB} = 1,\ 0 \le p_{vB} \le 1/4,\ 0 \le p_{wB} \le 1\}$. Therefore, the optimal solution spreads campaign B across viewer types v and w as much as possible, yielding $p_{vB} = 1/4$ and $p_{wB} = 3/4$. Now consider aggregating both viewer types into inventory block 1. The aggregate solution $\{p_{1A} = 9/4,\ p_{1B} = 3/4\}$ is equivalent to the disaggregate solution $\{p_{vA} = 3/4,\ p_{vB} = 1/4,\ p_{wB} = 3/4\}$, as can be seen from the following substitution. Yields are $y_v = \min(1, 1/(p_{1A} + p_{1B})) = 1/3$ and $y_w = \min(1, 1/p_{1B}) = 1$; hence $p_{vA} = y_v p_{1A} = (1/3)(9/4) = 3/4$, $p_{vB} = y_v p_{1B} = (1/3)(3/4) = 1/4$, and $p_{wB} = y_w p_{1B} = (1)(3/4) = 3/4$.*

Denote the optimal values of (OPP) and (APP) as z^{OPP} and z^{APP} respectively.

PROPOSITION 2. *(APP) is a restriction of (OPP). Hence, any point that is feasible in (APP) is feasible in (OPP). Thus, $z^{OPP} \le z^{APP}$.*

PROPOSITION 3. *The supply constraints $\sum_{k \in K_v} p_{vk} \le 1\ \forall v \in V$ are redundant in (APP).*

Proof. Notice that

$$\begin{aligned}\sum_{k \in K_v} p_{vk} &= \sum_{k \in K_v} y_v p_{i(v),k} = \sum_{k \in K_v} \min\left(1, 1/\sum_{k' \in K_v} p_{i(v),k'}\right) p_{i(v),k} \\ &= \min\left(1, 1/\sum_{k \in K_v} p_{i(v),k}\right) \sum_{k \in K_v} p_{i(v),k} = \min\left(\sum_{k \in K_v} p_{i(v),k}, 1\right) \le 1. \quad \square\end{aligned}$$

We introduce the quantities $a_{ik} := \sum_{v \in V_{ik}} s_v y_v^2$, $b_{ik} := \sum_{v \in V_{ik}} s_v y_v$, and $c_{ik} := a_{ik}/b_{ik}^2$ so that we may represent (APP) succinctly. The quantity c_{ik} is a measure of *yield variability* in the viewer types of inventory block i that campaign k targets. To see this, let Y_{ik} be a random variable that takes value y_v with probability s_v/s_{ik} for all $v \in V_{ik}$. Then b_{ik}/s_{ik} and a_{ik}/s_{ik} are the first and

second moments of Y_{ik}, respectively. A useful result, which follows directly from the fact that $\text{Var}[Y_{ik}] = \mathbb{E}[Y_{ik}^2] - \mathbb{E}[Y_{ik}]^2 \geq 0$, is that $c_{ik} \geq 1/s_{ik}$. Note that if all yields are 100% (i.e. $y_v = 1 \, \forall v \in V_{ik}$), then $a_{ik} = b_{ik} = s_{ik}$, and $c_{ik} = 1/s_{ik}$.

Thus, we can write the number of impressions planned for campaign k in inventory block i as

$$x_{ik} = \sum_{v \in V_{ik}} x_{vk} = \sum_{v \in V_{ik}} s_v p_{vk} = \sum_{v \in V_{ik}} s_v y_v p_{i(v),k} = p_{ik} \sum_{v \in V_{ik}} s_v y_v = b_{ik} p_{ik},$$

the objective function as

$$\sum_{\substack{k \in K, \\ v \in V_k}} c_v x_{vk}^2 = \sum_{\substack{k \in K, \\ v \in V_k}} s_v p_{vk}^2 = \sum_{\substack{k \in K, \\ i \in I_k}} \sum_{v \in V_{ik}} s_v (y_v p_{i(v),k})^2 = \sum_{\substack{k \in K, \\ i \in I_k}} p_{ik}^2 \sum_{v \in V_{ik}} s_v y_v^2 = \sum_{\substack{k \in K, \\ i \in I_k}} a_{ik} p_{ik}^2 = \sum_{\substack{k \in K, \\ i \in I_k}} c_{ik} x_{ik}^2,$$

and the impression goal constraint as

$$\sum_{v \in V_k} x_{vk} = \sum_{i \in I_k} \sum_{v \in V_{ik}} x_{vk} = \sum_{i \in I_k} x_{ik} = g_k.$$

Therefore, as with (OPP), there are two equivalent formulations for (APP): The *proportion formulation* (APP^{PROP}), which has decision variables p_{ik}, a_{ik}, and b_{ik}, and the *impression formulation* (APP^{IMP}), which has decision variables x_{ik} and c_{ik}:

$$\begin{array}{lll}
(APP^{PROP}) \text{ min} & \sum_{k \in K, i \in I_k} a_{ik} p_{ik}^2 & \\
\text{s.t.} & \sum_{i \in I_k} b_{ik} p_{ik} = g_k \; \forall \, k \in K & \text{(impression goals)} \\
& p_{ik} \geq 0 \quad \forall \, k \in K, \, i \in I_k & \text{(nonnegativity)} \\
& (\mathbf{p}, \mathbf{a}, \mathbf{b}) \in Y^{PROP} & \text{(nonlinear yield constraints)}
\end{array}$$

where Y^{PROP} is the set of points $\{p_{ik}, a_{ik}, b_{ik} \, \forall k \in K, i \in I_k\}$ that can be extended to a solution of the system

$$\begin{array}{ll}
y_v - \min\left(1, 1/\sum_{k \in K_v} p_{i(v),k}\right) = 0 \; \forall v \in V \\
a_{ik} - \sum_{v \in V_{ik}} s_v y_v^2 = 0 \; \forall k \in K, i \in I_k \\
b_{ik} - \sum_{v \in V_{ik}} s_v y_v = 0 \; \forall k \in K, i \in I_k
\end{array}$$

by picking appropriate values for the variables $y_v \, \forall v \in V$. Similarly, the impression formulation is

$$\begin{array}{lll}
(APP^{IMP}) \text{ min} & \sum_{k \in K, i \in I_k} c_{ik} x_{ik}^2 & \\
\text{s.t.} & \sum_{i \in I_k} x_{ik} = g_k \; \forall k \in K & \text{(impression goals)} \\
& x_{ik} \geq 0 \quad \forall k \in K, i \in I_k & \text{(nonnegativity)} \\
& (\mathbf{x}, \mathbf{c}) \in Y^{IMP} & \text{(nonlinear yield constraints)}
\end{array}$$

where Y^{IMP} is the set of points $\{x_{ik}, c_{ik}\, \forall k \in K, i \in I_k\}$ that can be extended to a solution of the system

$$\begin{aligned}
a_{ik} - \textstyle\sum_{v \in V_{ik}} s_v y_v^2 &= 0 \;\forall k \in K, i \in I_k \\
b_{ik} - \textstyle\sum_{v \in V_{ik}} s_v y_v &= 0 \;\forall k \in K, i \in I_k \\
c_{ik} - a_{ik}/b_{ik}^2 &= 0 \;\forall k \in K, i \in I_k \\
y_v - \min\left(1, 1/\textstyle\sum_{k \in K_v} p_{i(v),k}\right) &= 0 \;\forall v \in V \\
x_{ik} - b_{ik} p_{ik} &= 0 \;\forall k \in K, i \in I_k
\end{aligned}$$

by picking appropriate values for the variables $y_v\, \forall v \in V$ and $(p_{ik}, a_{ik}, b_{ik})\, \forall k \in K, i \in I_k$.

Although the supply constraints $\sum_{k \in K_v} p_{i(v),k} \leq 1\, \forall v \in V$ are not explicitly written, they are in fact satisfied, due to Proposition 3. As well, we can think of (APP) as having solutions in *both* the inventory block space and the viewer type space, since the disaggregation formula $p_{vk} = y_v p_{i(v),k}$ can be used to express the solution in the viewer type space.

6.2. Solution Approaches

There are two main problems that we consider: solving (APP) with a fixed inventory block partition, and solving (APP) while simultaneously refining the inventory block partition. For the case in which the partition is fixed, we give a heuristic that either finds a feasible solution with bounded distance from the (OPP) optimum or an infeasible solution with measurable shortfall for each impression goal. For the case where we are allowed to refine the inventory block partition, we give an algorithm that always terminates with an optimal solution to (OPP).

6.2.1. Fixed Inventory Block Partition. Management may insist on using a specific inventory block partition if, for example, the partition for planning ad server execution must coincide with the partition used by the sales team to price and bundle ad inventory. Using a fixed partition may also be desirable if the most accurate method for estimating viewers' arrival rates depends on a specific partition; e.g., see Agarwal et al. (2007), which describes a method to estimate impression counts using a prior generated from existing data, which essentially includes a given partition.

We make use of the following family of linear programs, parameterized by the objective coefficients $\mathbf{c} = \{c_{ik} : k \in K, i \in I_k\}$. We call this the *Auxiliary Transportation Problem*:

$$
\begin{array}{lll}
(AUX(\mathbf{c})) \min & \sum_{k\in K, i\in I_k} c_{ik} x_{ik}^2 & \\
\text{s.t.} & \sum_{i\in I_k} x_{ik} & = g_k \;\; \forall k \in K \qquad \text{(impression goals)} \\
& \sum_{k\in K_i} x_{ik} & \leq s_i \;\; \forall i \in I \qquad \text{(supply constraints)} \\
0 \leq & x_{ik} & \leq s_{ik} \;\; \forall k \in K, i \in I_k \quad \text{(arc flow bounds)}
\end{array}
$$

Note that $(AUX(\mathbf{c}))$ is equivalent to (APP^{IMP}) with objective coefficients fixed, nonlinear yield constraints dropped, supply constraints added, and variable upper bounds introduced. Although the supply constraints and upper bounds are redundant in (APP^{IMP}), they are not in $(AUX(\mathbf{c}))$.

Proposition 4. *The supply constraints and upper bounds present in* $(AUX(\mathbf{c}))$ *can be derived from* (OPP)*, thereby proving their validity.*

Proof. The supply constraints are aggregated from (OPP^{IMP}) as follows:

$$\sum_{k\in K_v} x_{vk} \leq s_v \implies \sum_{v\in V_i} \sum_{k\in K_v} x_{vk} \leq \sum_{v\in V_i} s_v \implies \sum_{k\in K_i} \sum_{v\in V_{ik}} x_{vk} \leq s_i \implies \sum_{k\in K_i} x_{ik} \leq s_i.$$

The upper bound on x_{ik} is derived from (OPP^{IMP}) as follows:

$$x_{vk} \leq s_v \implies \sum_{v\in V_{ik}} x_{vk} \leq \sum_{v\in V_{ik}} s_v \implies x_{ik} \leq s_{ik}. \quad \square$$

Our heuristic for finding a feasible solution to (APP), called GetCloseAndScaleUp, begins by *assuming* all yields are 100%; i.e. $y_v = 1 \; \forall v \in V$. Therefore, $a_{ik} = b_{ik} = s_{ik} \; \forall k \in K, i \in I_k$ and $c_{ik} = 1/s_{ik} \; \forall k \in K, i \in I_k$. With slight abuse of notation, we define $(AUX(1/\mathbf{s}))$ as the problem $(AUX(\mathbf{c}))$ with $c_{ik} = 1/s_{ik} \; \forall k \in K, i \in I_k$, and solve $(AUX(1/\mathbf{s}))$ to get an impression allocation $\mathbf{x}^{AUX(1/\mathbf{s})}$ which is "close" to optimal for (OPP), as well as corresponding inventory block weights $p_{ik}^0 = x_{ik}^{AUX(1/\mathbf{s})}/b_{ik}$. Scaling (Algorithm 1: ScaleInvBlockWeights) is then used to successively increase the inventory block weights p_{ik}^j at each iteration j until they hopefully converge to a feasible solution to (APP). The inputs for ScaleInvBlockWeights are the inventory block weights $\mathbf{p}^0$ from $(AUX(1/\mathbf{s}))$, as well as a threshold value ψ that limits the magnitude any element of $\mathbf{p}^j$ can take before the algorithm concludes that $\mathbf{p}^j$ is not converging to a feasible solution.

ScaleInvBlockWeights terminates with $p_{ik}^j = f_k^{CUM} p_{ik}^0$, where $f_k^{CUM} = f_k^{j-1} f_k^{j-2} \cdots f_k^1 f_k^0 \geq 1$. When $\mathbf{p}^j$ is feasible, the interpretation of f_k^{CUM} is the following: p_{ik}^0 is the correct inventory block

Algorithm 1 ScaleInvBlockWeights

Input(s): $\mathbf{p}^0$, ψ
Output: $\mathbf{p}^j$

1: Initialize the iteration counter at $j = 0$
2: Compute yields $\overline{y}_v^j := \min\left(1, 1/\sum_{k \in K_v} p_{i(v),k}^j\right)\ \forall v \in V$
3: Compute $\overline{b}_{ik}^j := \sum_{v \in V_{ik}} s_v \overline{y}_v^j\ \forall k \in K, i \in I_k$
4: Compute scaling factors $f_k^j := g_k / \sum_{i \in I_k} \overline{b}_{ik}^j p_{ik}^j\ \forall k \in K$
5: **if** $f_k^j = 1\ \forall k \in K$ **then**
6: $\quad$ $\mathbf{p}^j$ is feasible in (APP^{PROP})
7: $\quad$ **return** $\mathbf{p}^j$
8: **else if** $||\mathbf{p}^j||_\infty > \psi$ (i.e. $\mathbf{p}^j$ is not converging to a feasible solution) **then**
9: $\quad$ $\mathbf{p}^j$ is infeasible in (APP^{PROP})
10: $\quad$ **return** $\mathbf{p}^j$
11: **else**
12: $\quad$ Set $p_{ik}^{j+1} := f_k^j p_{ik}^j\ \forall k \in K, i \in I_k$; set $j := j + 1$; and go to step 2
13: **end if**

weight to use *if* all yields end up being 100%; but since yields are often lower, inventory block weights must be increased by a factor of f_k^{CUM} to compensate.

Note that $\sum_{i \in I_k} \overline{b}_{ik}^j p_{ik}^j = \sum_{i \in I_k} \sum_{v \in V_{ik}} s_v \overline{y}_v^j p_{ik}^j = \sum_{v \in V_k} s_v p_{vk}^j = \sum_{v \in V_k} x_{vk}^j$ is the total number of impressions allocated to campaign k at iteration j. Therefore, when ScaleInvBlockWeights terminates, we have $\Delta_k = g_k - \sum_{i \in I_k} \overline{b}_{ik} p_{ik}^j \geq 0$ as the number of impressions that campaign k is underallocated. When $\mathbf{p}^j$ is infeasible, $\Delta_k > 0$ for at least one campaign k. When this happens, management can either choose to execute this plan as-is (i.e. accept some reduction in impression goals) or refine the partition to recover feasibility, as we do in §6.2.2.

Other variants of GetCloseAndScaleUp are also reasonable to consider. In general, this class of iterative algorithm guesses yields $\mathbf{y}$, computes yield-dependent parameters $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, solves $(AUX(\mathbf{c}))$, evaluates the *actual yields* $\overline{\mathbf{y}}$ and *actual yield-dependent parameters* $(\overline{\mathbf{a}}, \overline{\mathbf{b}}, \overline{\mathbf{c}})$, adjusts $(\mathbf{a}, \mathbf{b}, \mathbf{c})$, and iterates, re-solving $(AUX(\mathbf{c}))$ until an acceptable solution is found. Note that GetCloseAndScaleUp is the special case with $b_{ik}^{j+1} = b_{ik}^j / f_k^j$; $a_{ik}^{j+1} = a_{ik}^j / (f_k^j)^2$; $c_{ik}^{j+1} = c_{ik}^j = 1/s_{ik}$.

Regardless of *how* a feasible solution to (APP) is found, we can always bound its distance from optimality using the optimal value from $(AUX(1/\mathbf{s}))$, as we will now show. Let $z^{AUX(1/\mathbf{s})}$ denote the optimal value of $(AUX(1/\mathbf{s}))$ and z^{OPP} denote the optimal value of (OPP).

Theorem 2. $z^{AUX(1/\mathbf{s})} \leq z^{OPP}$.

Proof. Any optimal dual solution for $(AUX(1/\mathbf{s}))$ is dual feasible in (OPP) with dual objective value $z^{AUX(1/\mathbf{s})}$. Since the dual problem of (OPP) is a maximization problem, $z^{OPP} \geq z^{AUX(1/\mathbf{s})}$. See §EC.5.1 for the full proof. □

COROLLARY 1. *Let z^{FEAS} be the value of a feasible solution to (OPP). Then the optimality gap is bounded: $z^{FEAS} - z^{OPP} \leq z^{FEAS} - z^{AUX(1/\mathbf{s})}$.*

Corollary 1 is important, because any feasible solution to (APP) is feasible for (OPP), and hence given any feasible aggregate solution, we have a bound on suboptimality that can be computed without solving the disaggregate (original) problem.

6.2.2. Partition Refinement. If we are allowed to refine the inventory block partition during the solution process, we can *always* find an optimal solution to (OPP) by using Algorithm 2: REFINEPARTITIONANDSOLVE, which successively creates new inventory blocks for groups of viewer types that are overallocated (i.e. have $\sum_{k \in K_v} p_{i(v),k} > 1$). We initialize the partition with a single inventory block which contains all viewer types; alternatively, we could of course begin with any inventory block partition and run the algorithm from that point forward. We have assumed that (OPP) is feasible; if it is not, REFINEPARTITIONANDSOLVE will detect infeasibility of (OPP) at Step 4, since eventually $(AUX(1/\mathbf{s}))$ will become infeasible. Note also that we don't need to run REFINEPARTITIONANDSOLVE to completion; we can always stop early, fix the inventory block partition, and run GETCLOSEANDSCALEUP to get a near-optimal or near-feasible solution.

THEOREM 3. *The solution $\mathbf{p}^j$ returned by* REFINEPARTITIONANDSOLVE *is optimal in (OPP).*

Proof. See §EC.5.2 . □

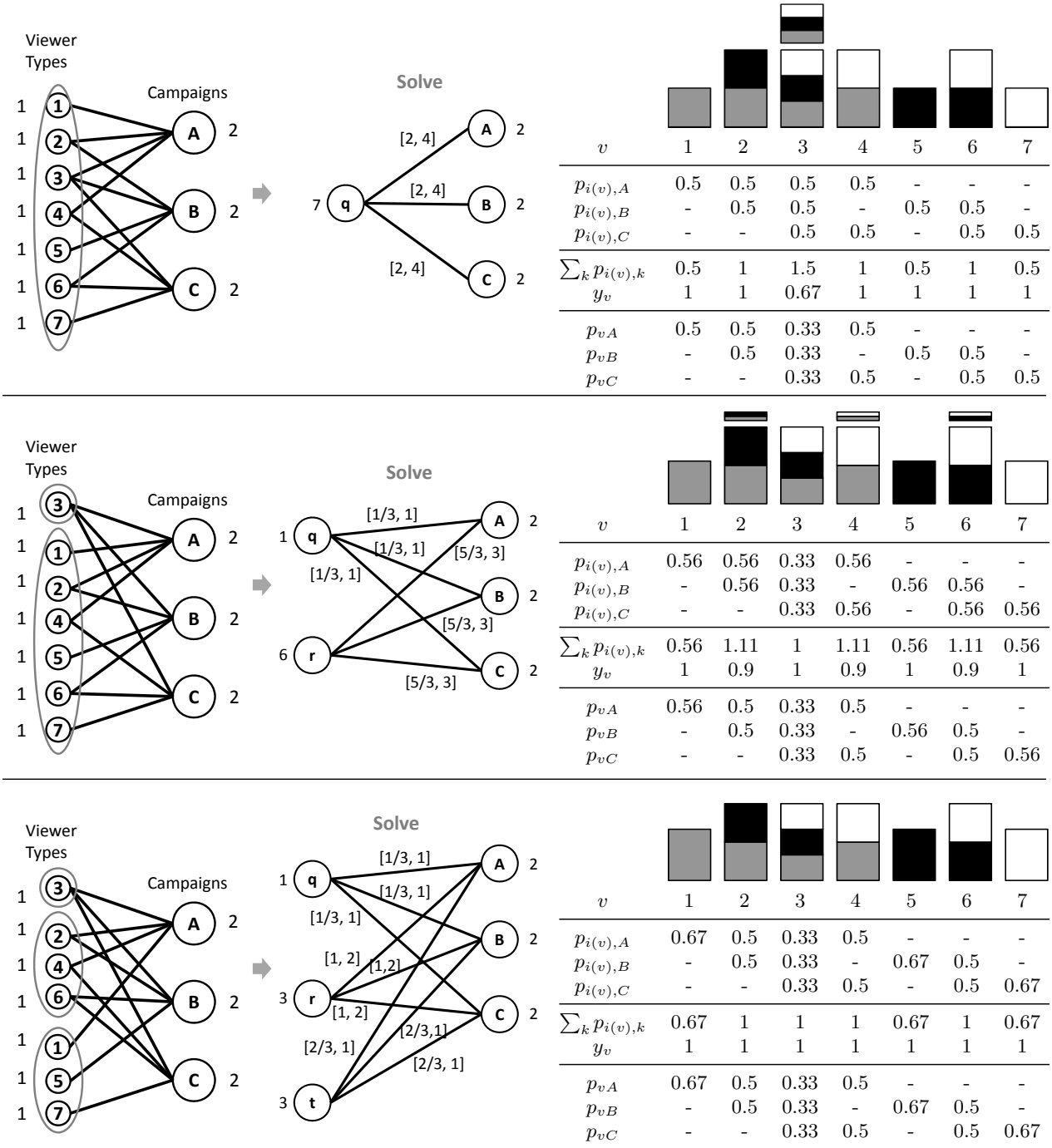
Figure 4 illustrates the progression of REFINEPARTITIONANDSOLVE on an (OPP) instance that has 7 viewer types with expected supply of 1, and 3 campaigns with impression goals of 2. Iteration 1 (top panel) begins by aggregating all viewer types to the same inventory block (q), which has expected supply of 7. Arcs in the aggregated network have capacities inferred from the maximum possible flow in the original network (e.g., campaign A is incident to 4 viewer types of expected supply 1; hence, aggregate arc (q, A) has a capacity of 4). Arcs in the aggregate network are labelled

Algorithm 2 RefinePartitionAndSolve

Initialize the iteration counter at $j = 0$
Initialize the partition to one big inventory block: $I^0 := \{1\}$, $V_1^0 := V$; i.e. $i(v) = 1 \; \forall v \in V$
loop
    Solve $(AUX(1/\mathbf{s}))$ with partition I^j to get $\mathbf{x}^j$
    Compute inventory block weights $p_{ik}^j := x_{ik}^j / s_{ik} \; \forall k \in K, i \in I_k^j$
    Compute yields $\overline{y}_v^j := \min\left(1, 1/\sum_{k \in K_v} p_{i(v),k}^j\right) \; \forall v \in V$
    For each $i \in I^j$, find the set of overallocated viewer types: $\widehat{V}_i^j := \{v \in V_i^j \mid \overline{y}_v^j < 1\}$
    Let $n^j := |i \in I^j \text{ s.t. } \widehat{V}_i^j \neq \varnothing|$ be the number of inventory blocks with overallocated viewer types
    if $n^j = 0$ **then**
        $\mathbf{p}^j$ is optimal in (OPP)
        return $\mathbf{p}^j$
    else
        Set $I^{j+1} := I^j \cup \{|I^j| + 1, |I^j| + 2, ..., |I^j| + n^j\}$ and $m := |I^j| + 1$
        for all $i \in I^j$ **do**
            if $\widehat{V}_i^j = \varnothing$ **then**
                Keep inventory block i unchanged: $V_i^{j+1} := V_i^j$
            else
                Split inventory block i in two: $V_i^{j+1} := V_i^j \backslash \widehat{V}_i^j$ and $V_m^{j+1} := \widehat{V}_i^j$
                $m := m + 1$
            end if
        end for
    end if
    $j := j + 1$
end loop

with $[x_{ik}, s_{ik}]$; i.e. the optimal solution to $(AUX(1/\mathbf{s}))$ and arc capacity, respectively. The tables on the right side of Figure 4 illustrate how the aggregate solution $p_{ik} = x_{ik}/s_{ik}$ gets scaled by yields y_v to produce the disaggregate solution $p_{vk} = y_v p_{i(v),k}$. Grey, black, and white bars graphically depict the proportions of each viewer type allocated to campaigns A, B, and C respectively. We see that at iteration 1, viewer type 3 is overallocated (i.e. $\sum_k p_{i(3),k} = 1.5 > 1$); the extent of which is depicted by the second set of bars drawn above viewer type 3. This overallocation induces an impression shortfall $\Delta_k = g_k - \sum_{v \in V_k} s_v p_{vk} = 2 - 1.83 = 0.17$ for all campaigns k. Since viewer type 3 is the sole overallocated viewer type, it becomes its own inventory block in the subsequent iteration. Iteration 2 (middle panel) now solves an aggregate problem with viewer type 3 as inventory block q and viewer types $\{1, 2, 4, 5, 6, 7\}$ as inventory block r. At this point, viewer types 2, 4, and 6 are overallocated, with impression shortfall $\Delta_k = 0.11$ for all campaigns. Iteration 3 (bottom panel) modifies the previous aggregation by splintering off viewer types $\{2, 4, 6\}$ into their own inventory

block (the new r, with viewer types $\{1,5,7\}$ becoming t). Finally, iteration 3 terminates with no viewer types overallocated, and by Theorem 3 we have found an optimal solution to (OPP).



v	1	2	3	4	5	6	7
$p_{i(v),A}$	0.5	0.5	0.5	0.5	-	-	-
$p_{i(v),B}$	-	0.5	0.5	-	0.5	0.5	-
$p_{i(v),C}$	-	-	0.5	0.5	-	0.5	0.5
$\sum_k p_{i(v),k}$	0.5	1	1.5	1	0.5	1	0.5
y_v	1	1	0.67	1	1	1	1
p_{vA}	0.5	0.5	0.33	0.5	-	-	-
p_{vB}	-	0.5	0.33	-	0.5	0.5	-
p_{vC}	-	-	0.33	0.5	-	0.5	0.5

v	1	2	3	4	5	6	7
$p_{i(v),A}$	0.56	0.56	0.33	0.56	-	-	-
$p_{i(v),B}$	-	0.56	0.33	-	0.56	0.56	-
$p_{i(v),C}$	-	-	0.33	0.56	-	0.56	0.56
$\sum_k p_{i(v),k}$	0.56	1.11	1	1.11	0.56	1.11	0.56
y_v	1	0.9	1	0.9	1	0.9	1
p_{vA}	0.56	0.5	0.33	0.5	-	-	-
p_{vB}	-	0.5	0.33	-	0.56	0.5	-
p_{vC}	-	-	0.33	0.5	-	0.5	0.56

v	1	2	3	4	5	6	7
$p_{i(v),A}$	0.67	0.5	0.33	0.5	-	-	-
$p_{i(v),B}$	-	0.5	0.33	-	0.67	0.5	-
$p_{i(v),C}$	-	-	0.33	0.5	-	0.5	0.67
$\sum_k p_{i(v),k}$	0.67	1	1	1	0.67	1	0.67
y_v	1	1	1	1	1	1	1
p_{vA}	0.67	0.5	0.33	0.5	-	-	-
p_{vB}	-	0.5	0.33	-	0.67	0.5	-
p_{vC}	-	-	0.33	0.5	-	0.5	0.67

Figure 4 RefinePartitionAndSolve takes 3 iterations to solve this instance of (OPP) to optimality.

6.3. Computational Results

On a structurally rich set of randomly generated instances, algorithms GetCloseAndScaleUp and RefinePartitionAndSolve efficiently find small, aggregated, near-optimal solutions to (OPP), thereby confirming their practicability. We generate each instance as follows: First, each campaign and each viewer type are assigned a link intensity, and with probability (campaign link intensity) $\times$ (viewer type link intensity), a link between campaign and viewer type is created in the underlying transportation network, thereby defining the instance's targeting. Next, expected supply for each viewer type is randomly generated, and viewer types that target the same set of campaigns are merged (summing together expected supplies). Finally, impression goals are generated by iterating through the list of campaigns and recording "reservations" from each campaign. A campaign will attempt to reserve the same proportion of supply from each viewer type it targets, settling for less if insufficient supply remains.

Our testing makes use of six test cases, which, in addition to size, are distinguished by *sell-through* – the ratio of total impression goals to total supply – and *targeting percentage* – the average proportion of viewer types targeted by a campaign. Figure 5 lists the test cases along with the uniform distributions used to generate their link intensities and the reservation proportions used to generate their impression goals. In all cases, expected supply is the product of a Pareto random variable with minimum 0, mean 1, and shape parameter 5 (to model heavy-tail arrivals of viewers) and the fixed number 32 (a reasonable rate of impressions per arrival for video games). We use the Pareto distribution since in practice some viewer types yield a disproportionately large share of arrivals (see Agarwal et al. 2007); however, similar results are achieved with Uniform[0,2] arrivals.

The first three columns of Figure 6 illustrate the progression of RefinePartitionAndSolve on the "Large, Locally Tight" test case. Column 1 lists the number of refinement iterations taken, Column 2 the number of inventory blocks introduced, and Column 3 measures *feasibility*, defined as the proportion of impression goals met in expectation by the disaggregated plan. The results from ten randomly generated instances are summarized by reporting averages, as well as the 10^{th} and 90^{th} percentiles; e.g., after the 10^{th} iteration, on average 990.8 inventory blocks were introduced,

Test Case	# Campaigns	# Viewer Types	Sell-through	Targeting %	Campaign Link Intensity	Viewer Type Link Intensity	Reservation Proportion
Small, Loose	100	10,000	67%	18.75%	U[0.1, 0.4]	U[0.5, 1]	3.5%
Small, Globally Tight	100	10,000	92%	18.75%	U[0.1, 0.4]	U[0.5, 1]	6%
Small, Locally Tight	100	10,000	95%	3.9%	U[0.01, 0.12]	U[0.2, 1]	$*$
Large, Loose	100	100,000	63%	18.75%	U[0.1, 0.4]	U[0.5, 1]	3.5%
Large, Globally Tight	100	100,000	92%	18.75%	U[0.1, 0.4]	U[0.5, 1]	6%
Large, Locally Tight	100	100,000	94%	3.9%	U[0.01, 0.12]	U[0.2, 1]	$*$

Figure 5 The six test cases. In cases marked $*$, reservation proportions for each campaign were drawn from {40%, 90%, and 100%} with probabilities 0.6, 0.3, and 0.1, respectively. Globally tight cases have high sell-through, while locally tight cases have both high sell-through and low targeting percentage.

and 974.2 and 1004.2 were the 10^{th} and 90^{th} percentiles. We see RefinePartitionAndSolve performs very well: Starting from a single inventory block at iteration 0, by iteration 4 the viewer type space was intelligently aggregated into 16 inventory blocks to yield a 97.2%-feasible solution, and 99.9% feasibility was achieved at iteration 9 using on average 501.7 inventory blocks – a significant level of aggregation considering there are 100,000 viewer types.

The performance of GetCloseAndScaleUp is illustrated by Columns 4-7 of Figure 6: Column 4 reports the proportion of instances that *successfully terminated*; i.e. those where a 99.99%+ feasible plan was found within 100 scaling iterations and the scaling factors f_k^{CUM} did not exceed 100; Columns 5-7 report summary statistics for the *subset* of instances that terminated successfully. In particular, Column 5 lists the number of scaling iterations required to find a feasible solution, Column 6 bounds the optimality gap using Corollary 1, and Column 7 reports the actual optimality gap. We can see that GetCloseAndScaleUp performs very well: scaled solutions are often within a few percent of optimality even when the inventory block partition is highly aggregated.

Since the number of scaling iterations required to find a feasible solution generally drops as the instance is disaggregated, the fastest way to find a feasible plan often involves running several refinement iterations and then switching to the scaling algorithm. For example, in Figure 6 we can see that after 8 iterations of RefinePartitionAndSolve, on average 254.1 inventory blocks were introduced and a 99.8%-feasible solution was found. Running GetCloseAndScaleUp on this solution was successful in 100% of instances, and after an average of 18.2 scaling iterations, a feasible solution within 0.1% of both proven and actual optimality was found.

When we compare our algorithms' performance across the six test cases from Figure 5, we find

Refine Iter	Inventory Blocks	Feasibility	Scaling Success	Scaling Iters	Optimality Gap Bound	Optimality Gap Actual
0	$1.0^{[1.0,1.0]}$	$0.751^{[0.739,0.763]}$	0.0	–	–	–
1	$2.0^{[2.0,2.0]}$	$0.835^{[0.823,0.848]}$	0.3	$88.7^{[79.8,96.6]}$	$0.148^{[0.138,0.155]}$	$0.031^{[0.028,0.034]}$
2	$4.0^{[4.0,4.0]}$	$0.905^{[0.896,0.915]}$	0.5	$77.8^{[63.2,91.6]}$	$0.061^{[0.056,0.066]}$	$0.023^{[0.021,0.025]}$
3	$8.0^{[8.0,8.0]}$	$0.948^{[0.942,0.956]}$	0.6	$70.5^{[54.0,91.0]}$	$0.030^{[0.028,0.032]}$	$0.017^{[0.016,0.018]}$
4	$16.0^{[16.0,16.0]}$	$0.972^{[0.969,0.975]}$	0.8	$68.0^{[47.8,89.1]}$	$0.015^{[0.014,0.016]}$	$0.010^{[0.010,0.011]}$
5	$32.0^{[32.0,32.0]}$	$0.984^{[0.983,0.986]}$	0.9	$59.7^{[40.0,78.2]}$	$0.007^{[0.007,0.008]}$	$0.006^{[0.005,0.006]}$
6	$64.0^{[64.0,64.0]}$	$0.991^{[0.990,0.992]}$	1.0	$51.7^{[32.1,79.1]}$	$0.003^{[0.003,0.004]}$	$0.003^{[0.003,0.003]}$
7	$127.8^{[127.0,128.0]}$	$0.995^{[0.995,0.996]}$	1.0	$31.6^{[22.5,41.9]}$	$0.002^{[0.001,0.002]}$	$0.001^{[0.001,0.002]}$
8	$254.1^{[252.9,255.1]}$	$0.998^{[0.997,0.998]}$	1.0	$18.2^{[14.7,24.2]}$	$0.001^{[0.001,0.001]}$	$0.001^{[0.000,0.001]}$
9	$501.7^{[493.7,507.0]}$	$0.999^{[0.999,0.999]}$	1.0	$10.6^{[8.0,13.2]}$	$0.000^{[0.000,0.000]}$	$0.000^{[0.000,0.000]}$
10	$990.8^{[974.2,1004.2]}$	$0.999^{[0.999,0.999]}$	1.0	$6.0^{[5.0,7.2]}$	$0.000^{[0.000,0.000]}$	$0.000^{[0.000,0.000]}$

Figure 6 Performance of RefinePartitionAndSolve and GetCloseAndScaleUp on the "Large, Locally Tight" test case.

that to achieve a solution a given distance from optimality, 1) larger cases take longer than smaller ones but often require a comparable number of refinement iterations and inventory blocks, 2) locally tight cases need the most refinement and scaling iterations and produce the most inventory blocks, and 3) globally tight cases take the most time to solve since they are the largest to represent. In all cases, the algorithms performed very well. For the details, see §EC.6.

7. Conclusions

Among the media vehicles that can be considered *Guaranteed Targeted Display Advertising*, allocating impressions from campaigns to audience segments can be done with a transportation problem with quadratic objective. Models of audience uncertainty, forecast error, and the random slotting of the ad server were used to derive sufficient conditions for *when* the quadratic objective minimizes the variance of the number of impressions served and maximizes expected reach, so that ad managers can understand *if* and *why* their ad server is optimal with respect to variance and reach. In addition, we studied *aggregation* of the viewer type space and gave two algorithms to solve the original large planning problem: GetCloseAndScaleUp, which assumes a fixed partition of the viewer type space and finds a feasible solution with measurable optimality gap; and RefinePartitionAndSolve, which successively refines the partition at each iteration, terminating with an optimal solution if one exists.

Acknowledgments

Thanks to the associate editor, three anonymous referees, and to Alan Scheller-Wolf, Sridhar Tayur, and Mike Trick for their valuable feedback.

References

Abrams, Z., S. S. Keerthi, O. Mendelevitch, J. A. Tomlin. 2008. Ad Delivery with Budgeted Advertisers: A Comprehensive LP Approach. *Journal of Electronic Commerce Research* **9**(1).

Adler, M., P. B. Gibbons, Y. Matias. 2002. Scheduling space-sharing for internet advertising. *Journal of Scheduling* **5**(2) 103–119.

Agarwal, D., A. Z. Broder, D. Chakrabarti, D. Diklic, V. Josifovski, M. Sayyadian. 2007. Estimating rates of rare events at multiple resolutions. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM New York, NY, USA, 16–25.

Araman, V. F., K. Fridgeirsdottir. 2008. Online Advertising: Revenue Management Approach. Working paper. London Business School.

Arango, T. 2008. Cable firms join forces to attract focused ads. *The New York Times* March 10.

Birge, J. R. 1985. Aggregation bounds in stochastic linear programming. *Mathematical Programming* **31**(1) 25–41.

Chlebus, E., J. Brazier. 2007. Nonstationary Poisson modeling of web browsing session arrivals. *Information Processing Letters* **102**(5) 187–190.

Danaher, P. 2008. Advertising models. B. Wierenga, ed., *Handbook of marketing decision models*. Springer Verlag.

Dawande, M., S. Kumar, C. Sriskandarajah. 2003. Performance bounds of algorithms for scheduling advertisements on a web page. *Journal of Scheduling* **6**(4) 373–394.

De Reyck, B., Z. Degraeve. 2003. Broadcast scheduling for mobile advertising. *Operations Research* **51**(4) 509–517.

Edelman, B., M. Ostrovsky, M. Schwarz. 2007. Internet advertising and the generalized second-price auction: selling billions of dollars worth of keywords. *American Economic Review* **97**(1) 242–259.

Fridgeirsdottir, K., S. Najafi-Asadolahi. 2008. Revenue management for online advertising: impatient advertisers. Working paper. London Business School.

Ghosh, A., P. McAfee, K. Papineni, S. Vassilvitskii. 2009. Bidding for representative allocations for display advertising. *Internet and Network Economics* 208–219.

Hallerman, D. 2009. U. S. advertising spending: the new reality. *eMarketer* May 2009.

Interactive Advertising Bureau. 2009. 2008 IAB Internet Advertising Revenue Report. `http://www.iab.net/media/file/IAB_PwC_2008_full_year.pdf`.

Kumar, S., V. S. Jacob, C. Sriskandarajah. 2006. Scheduling advertisements on a web page to maximize revenue. *European Journal of Operational Research* **173**(3) 1067–1089.

Langheinrich, M., A. Nakamura, N. Abe, T. Kamba, Y. Koseki. 1999. Unintrusive customization techniques for Web advertising. *Computer Networks: The International Journal of Computer and Telecommunications Networking* **31**(11) 1259–1272.

Leisten, R. 1997. A posteriori error bounds in linear programming aggregation. *Computers and Operations Research* **24**(1) 1–16.

Litvinchev, I., V. Tsurkov. 2003. *Aggregation in large-scale optimization*. Kluwer Academic Publishers.

Litvinchev, I. S., S. Rangel. 2006. Using error bounds to compare aggregated generalized transportation models. *Annals of Operations Research* **146**(1) 119–134.

Liu, Z., N. Niclausse, C. Jalpa-Villanueva. 2001. Traffic model and performance evaluation of Web servers. *Performance Evaluation* **46**(2-3) 77–100.

Mandel, C. 2007. Eye-catching glance at the future. *Globe and Mail* June 28.

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

Menon, S., A. Amiri. 2004. Scheduling banner advertisements on the web. *INFORMS Journal on Computing* **16**(1).

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

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

Rogers, D. F., R. D. Plante, R. T. Wong, J. R. Evans. 1991. Aggregation and disaggregation techniques and methodology in optimization. *Operations Research* 553–582.

Surmanek, J. 1995. *Media planning: a practical guide*. McGraw-Hill.

Tomlin, J., D. Agarwal, J. Yang. 2008. A log-linear model for allocating overlapping inventory to on-line advertisers. Presented at INFORMS Annual Conference.

Tomlin, J., V. Bharadwaj, M. Saunders. 2009. Solution of a stochastic model for allocation of on-line advertising inventory. Presented at INFORMS Annual Conference.

Tomlin, J. A. 2000. An entropy approach to unintrusive targeted advertising on the Web. *Computer Networks* **33**(1-6) 767–774.

Turner, J. 2010. Ad slotting & pricing: new media planning models for new media. Ph.D. thesis, Tepper School of Business, Carnegie Mellon University.

Turner, J., A. Scheller-Wolf, S. Tayur. 2011. Scheduling of Dynamic In-Game Advertising. *Operations Research* **59**(1) 1–16.

Vakhutinsky, I. Y., L. M. Dudkin, A. A. Ryvkin. 1979. Iterative aggregation: a new approach to the solution of large-scale problems. *Econometrica* 821–841.

Walsh, W. E., C. Boutilier, T. Sandholm, R. Shields, G. Nemhauser, D. C. Parkes. 2010. Automated channel abstraction for advertising auctions. *Proceedings of the 24th AAAI Conference on Artificial Intelligence* .

Wright, S. E. 1994. Primal-dual aggregation and disaggregation for stochastic linear programs. *Mathematics of Operations Research* 893–908.

Yang, J., J. Tomlin. 2008. Advertising inventory allocation based on multi-objective optimization. Presented at INFORMS Annual Conference.

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

Zipkin, P. H. 1980a. Bounds for aggregating nodes in network problems. *Mathematical Programming* **19**(1) 155–177.

Zipkin, P. H. 1980b. Bounds for row-aggregation in linear programming. *Operations Research* 903–916.

Zipkin, P. H. 1980c. Bounds on the effect of aggregating variables in linear programs. *Operations Research* 403–418.

Zipkin, P. H. 1982. Aggregation and disaggregation in convex network problems. *Networks* **12**(2).

Zipkin, P. H., K. Raimer. 1983. An improved disaggregation method for transportation problems. *Mathematical Programming* **26**(2) 238–242.

The Planning of Guaranteed Targeted Display Advertising

John Turner
The Paul Merage School of Business, University of California at Irvine, Irvine, CA 92697-3125
john.turner@uci.edu

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

Key words: guaranteed targeted display advertising, advertising, planning, aggregation

---

Supplementary Material

EC.1. Derivations for Example 2

In §3.3 we presented a specific example of ad server execution, and claimed that $\mathbb{E}[F^r_{vk}] = p_{vk}$; $\alpha_v(p) = \frac{2\lfloor np\rfloor+1}{n}(p - \frac{\lfloor np\rfloor}{n}) + \frac{\lfloor np\rfloor^2}{n^2}$. These derivations are included here. For clarity, we take $p \equiv p_{vk}$.

$$\begin{aligned}
\mathbb{E}[F^r_{vk}] &= \frac{\lfloor np\rfloor}{n}(\lfloor np\rfloor + 1 - np) + \frac{\lceil np\rceil}{n}(np - \lfloor np\rfloor) \\
&= \frac{\lfloor np\rfloor}{n}(\lfloor np\rfloor + 1 - np) + \frac{\lfloor np\rfloor + 1}{n}(np - \lfloor np\rfloor) \\
&= \frac{\lfloor np\rfloor}{n}(\lfloor np\rfloor + 1 - np + np - \lfloor np\rfloor) + \frac{1}{n}(np - \lfloor np\rfloor) \\
&= \frac{\lfloor np\rfloor}{n}(1) + \frac{1}{n}(np - \lfloor np\rfloor) \\
&= \frac{\lfloor np\rfloor}{n} + p - \frac{\lfloor np\rfloor}{n} \\
&= p
\end{aligned}$$

$$\begin{aligned}
\alpha(p) := \mathbb{E}[(F^r_{vk})^2] &= \left(\frac{\lfloor np\rfloor}{n}\right)^2(\lfloor np\rfloor + 1 - np) + \left(\frac{\lfloor np\rfloor + 1}{n}\right)^2(np - \lfloor np\rfloor) \\
&= \left(\frac{\lfloor np\rfloor}{n}\right)^2(\lfloor np\rfloor + 1 - np) + \left(\left(\frac{\lfloor np\rfloor}{n}\right)^2 + 2\frac{\lfloor np\rfloor}{n^2} + \frac{1}{n^2}\right)(np - \lfloor np\rfloor) \\
&= \left(\frac{\lfloor np\rfloor}{n}\right)^2(\lfloor np\rfloor + 1 - np + np - \lfloor np\rfloor) + \left(2\frac{\lfloor np\rfloor}{n^2} + \frac{1}{n^2}\right)(np - \lfloor np\rfloor) \\
&= \left(\frac{\lfloor np\rfloor}{n}\right)^2(1) + \left(2\frac{\lfloor np\rfloor}{n^2} + \frac{1}{n^2}\right)(np - \lfloor np\rfloor) \\
&= \frac{\lfloor np\rfloor^2}{n^2} + 2\frac{\lfloor np\rfloor p}{n} - 2\frac{\lfloor np\rfloor^2}{n^2} + \frac{p}{n} - \frac{\lfloor np\rfloor}{n^2} \\
&= 2\frac{\lfloor np\rfloor p}{n} - \frac{\lfloor np\rfloor^2}{n^2} + \frac{p}{n} - \frac{\lfloor np\rfloor}{n^2} \\
&= \frac{2\lfloor np\rfloor + 1}{n}p - \frac{\lfloor np\rfloor^2 + \lfloor np\rfloor}{n^2} \\
&= \frac{2\lfloor np\rfloor + 1}{n}p - \frac{\lfloor np\rfloor(\lfloor np\rfloor + 1)}{n^2} \\
&= \frac{2\lfloor np\rfloor + 1}{n}(p - \frac{\lfloor np\rfloor}{n}) - \frac{\lfloor np\rfloor(\lfloor np\rfloor + 1)}{n^2} + \frac{2\lfloor np\rfloor + 1}{n}(\frac{\lfloor np\rfloor}{n}) \\
&= \frac{2\lfloor np\rfloor + 1}{n}(p - \frac{\lfloor np\rfloor}{n}) - \frac{\lfloor np\rfloor(\lfloor np\rfloor + 1)}{n^2} + \frac{2\lfloor np\rfloor^2}{n^2} + \frac{\lfloor np\rfloor}{n^2} \\
&= \frac{2\lfloor np\rfloor + 1}{n}(p - \frac{\lfloor np\rfloor}{n}) - \frac{\lfloor np\rfloor^2 + \lfloor np\rfloor - 2\lfloor np\rfloor^2 - \lfloor np\rfloor}{n^2} \\
&= \frac{2\lfloor np\rfloor + 1}{n}(p - \frac{\lfloor np\rfloor}{n}) + \frac{\lfloor np\rfloor^2}{n^2}
\end{aligned}$$

Furthermore, $\alpha(p) = \frac{2\lfloor np\rfloor+1}{n}(p - \frac{\lfloor np\rfloor}{n}) + \frac{\lfloor np\rfloor^2}{n^2}$ is convex in p.

Proof. We show that $\alpha(p)$ is piecewise-linear increasing (and hence convex). Let $\alpha_j(p) = \frac{2j+1}{n}(p-\frac{j}{n})+\frac{j^2}{n^2}$, $j \in \{0..n-1\}$. Then $\alpha(p) = \alpha_j(p)$ when $p \in \left[\frac{j}{n}, \frac{j+1}{n}\right)$. Each segment $\alpha_j(p)$ is linear in p. As well, $\alpha(p)$ is continuous: at the point $p = \frac{j+1}{n}$ between segments j and $j+1$, we have $\alpha(\frac{j+1}{n}) = \lim_{p\uparrow\frac{j+1}{n}} \alpha_j(p) = \lim_{p\downarrow\frac{j+1}{n}} \alpha_{j+1}(p) = \frac{(j+1)^2}{n^2}$, since

$$\begin{aligned}
\lim_{p\uparrow\frac{j+1}{n}} \alpha_j(p) &= \lim_{p\uparrow\frac{j+1}{n}} \left(\frac{2j+1}{n}\right)\left(p-\frac{j}{n}\right)+\frac{j^2}{n^2} \\
&= \left(\frac{2j+1}{n}\right)\left(\frac{j+1}{n}-\frac{j}{n}\right)+\frac{j^2}{n^2} \\
&= \left(\frac{2j+1}{n}\right)\left(\frac{1}{n}\right)+\frac{j^2}{n^2} \\
&= \frac{2j+1+j^2}{n^2} \\
&= \frac{(j+1)^2}{n^2} \\
&= \left(\frac{2(j+1)+1}{n}\right)\left(\frac{j+1}{n}-\frac{j+1}{n}\right)+\frac{(j+1)^2}{n^2} \\
&= \lim_{p\downarrow\frac{j+1}{n}} \left(\frac{2(j+1)+1}{n}\right)\left(p-\frac{j+1}{n}\right)+\frac{(j+1)^2}{n^2} \\
&= \lim_{p\downarrow\frac{j+1}{n}} \alpha_{j+1}(p).
\end{aligned}$$

Finally, the slope of segment j is $\frac{2j+1}{n}$; thus $\alpha(p)$ is increasing in p. □

EC.2. Derivations for the Mean and Variance of X_{vk}

We require the following technical lemmas:

LEMMA EC.1. *If $Y = \sum_{n=1..N} X_n$ is the sum of a random number of i.i.d. random variables, where the X_n's and N are mutually independent, then from first principles we have $\mathbb{E}[Y] = \mathbb{E}[N]\mathbb{E}[X]$ and $\mathrm{Var}[Y] = \mathbb{E}[N]\mathbb{E}[X^2] + \mathbb{E}[X]^2(\mathrm{Var}[N] - \mathbb{E}[N])$.*

LEMMA EC.2. *If $M \sim$ Poisson(L), where L is a random variable, then $\mathrm{Var}[M] = \mathrm{Var}[L] + \mathbb{E}[L]$.*

Proof.

$$\begin{aligned}
\mathrm{Var}[M] &= \mathrm{Var}[\mathbb{E}[M|L]] + \mathbb{E}[\mathrm{Var}[M|L]];\ \text{by the law of total variance} \\
&= \mathrm{Var}[L] + \mathbb{E}[L];\ \text{since } \mathbb{E}[M|L] = \mathrm{Var}[M|L] = L \text{ for } M \sim \mathrm{Poisson}(L). \quad \square
\end{aligned}$$

These quantities are used in the main derivations:

LEMMA EC.3. $\mathbb{E}[F^r_{vk}Y^r_v] = \mu_v p_{vk}$.

Proof. $\mathbb{E}[F^r_{vk}] = p_{vk}$, $\mathbb{E}[Y^r_v] = \mu_v$, and $\mathbb{E}[F^r_{vk}Y^r_v] = \mathbb{E}[F^r_{vk}]\mathbb{E}[Y^r_v]$ since F^r_{vk} and Y^r_v are independent. □

LEMMA EC.4. $\mathbb{E}[(F^r_{vk}Y^r_v)^2] = (\sigma_v^2 + \mu_v^2)\alpha_v(p_{vk})$.

Proof. $\mathbb{E}[(F^r_{vk})^2] = \alpha_v(p_{vk})$, $\mathbb{E}[(Y^r_v)^2] = \sigma_v^2 + \mu_v^2$, and $\mathbb{E}[(F^r_{vk}Y^r_v)^2] = \mathbb{E}[(F^r_{vk})^2]\mathbb{E}[(Y^r_v)^2]$ since F^r_{vk} and Y^r_v are independent. □

We now derive the mean and variance of the number of impressions served to campaign k from viewer type v.

LEMMA EC.5. *The expected number of impressions served to campaign k from viewer type v is* $\mathbb{E}[X_{vk}] = \lambda_v \mu_v p_{vk} = s_v p_{vk}$.

Proof. Taking the expectation of Equation (7), we get:

$$\begin{aligned}
\mathbb{E}[X_{vk}] &= \mathbb{E}\left[\sum_{r=1..M_v} F^r_{vk}Y^r_v\right] \\
&= \mathbb{E}[M_v]\mathbb{E}[F_{vk}Y^r_v] \text{ by Lemma EC.1} \\
&= \mathbb{E}[M_v]\mu_v p_{vk} \text{ by Lemma EC.3} \\
&= \lambda_v \mu_v p_{vk} = s_v p_{vk} \text{ by iterating the expectation } \mathbb{E}[M_v] = \mathbb{E}[\Lambda_v] = \lambda_v \ . \quad \square
\end{aligned}$$

LEMMA EC.6. *The variance of the number of impressions served to campaign k from viewer type v is* $\mathrm{Var}[X_{vk}] = (\sigma_v^2 + \mu_v^2)\lambda_v \alpha_v(p_{vk}) + \mu_v^2 p_{vk}^2 \mathrm{Var}[\Lambda_v]$.

Proof. Taking the variance of Equation (7), we get:

$$\begin{aligned}
\mathrm{Var}[X_{vk}] &= \mathrm{Var}\left[\sum_{i=1..M_v} F^r_{vk}Y^r_v\right] \\
&= \mathbb{E}[M_v]\mathbb{E}[(F^r_{vk}Y^r_v)^2] + \mathbb{E}[F^r_{vk}Y^r_v]^2 \left(\mathrm{Var}[M_v] - \mathbb{E}[M_v]\right) \text{ by Lemma EC.1} \\
&= \mathbb{E}[M_v]\mathbb{E}[(F^r_{vk}Y^r_v)^2] + \mathbb{E}[F^r_{vk}Y^r_v]^2 \left(\mathrm{Var}[\Lambda_v] + \mathbb{E}[\Lambda_v] - \mathbb{E}[M_v]\right) \text{ by Lemma EC.2} \\
&= \mathbb{E}[\Lambda_v]\mathbb{E}[(F^r_{vk}Y^r_v)^2] + \mathbb{E}[F^r_{vk}Y^r_v]^2 \left(\mathrm{Var}[\Lambda_v] + \mathbb{E}[\Lambda_v] - \mathbb{E}[\Lambda_v]\right) \\
&= \lambda_v \mathbb{E}[(F^r_{vk}Y^r_v)^2] + \mathbb{E}[F^r_{vk}Y^r_v]^2 \mathrm{Var}[\Lambda_v] \\
&= \lambda_v(\sigma_v^2 + \mu_v^2)\alpha_v(p_{vk}) + \mu_v^2 p_{vk}^2 \mathrm{Var}[\Lambda_v] \text{ by Lemmas EC.3 and EC.4.} \quad \square
\end{aligned}$$

EC.3. Optimality Results for the Equal-Proportion Allocation

THEOREM 1. *Consider the objective function* $f(\mathbf{p}) = \sum_{k\in K, v\in V_k} s_v h(p_{vk})$*, where* $h : \mathbb{R} \to \mathbb{R}$ *is convex (but possibly nondifferentiable). The equal-proportion allocation* $\mathbf{p} = \mathbf{q}$ *is optimal for the problem:*

$$\begin{aligned} (P1)\ \min\ & f(\mathbf{p}) \\ s.t.\ & \textstyle\sum_{v\in V_k} s_v p_{vk} = g_k \quad \forall k \in K && \textit{(impression goals)} \\ & p_{vk} \geq 0 \qquad\qquad \forall k \in K, \forall v \in V_k && \textit{(non-negativity)} \end{aligned}$$

Proof. Problem $(P1)$ decomposes into k single-campaign planning problems of the form

$$\begin{aligned} (P1k)\ \min\ & \textstyle\sum_{v\in V_k} s_v h(p_{vk}) \\ \text{s.t.}\ & \textstyle\sum_{v\in V_k} s_v p_{vk} = g_k && \text{(impression goal)} \\ & p_{vk} \geq 0 \qquad\qquad \forall v \in V_k && \text{(non-negativity)} \end{aligned}$$

Since $h(\cdot)$ is convex, by Jensen's Inquality

$$h\left(\frac{\sum_{v\in V_k} s_v p_{vk}}{\sum_{v\in V_k} s_v}\right) \leq \frac{\sum_{v\in V_k} s_v h(p_{vk})}{\sum_{v\in V_k} s_v}$$

holds for any feasible solution $\{p_{vk}\}_{v\in V_k}$ to $(P1k)$. Since $q_k = g_k / \sum_{v\in V_k} s_v = \sum_{v\in V_k} s_v p_{vk} / \sum_{v\in V_k} s_v$, we have $h(q_k) \leq \sum_{v\in V_k} s_v p_{vk} / \sum_{v\in V_k} s_v$. Hence, $\sum_{v\in V_k} s_v h(q_k) \leq \sum_{v\in V_k} s_v p_{vk}$ holds for any feasible $\{p_{vk}\}_{v\in V_k}$. Therefore, the equal-proportion allocation $\{q_k, \cdots, q_k\}$ is a minimizer of $(P1k)$, and by extension $\mathbf{q}$ minimizes $(P1)$. □

EC.3.1. Corollaries

We show that under certain conditions, the expressions for expected reach, stochastic variance, and forecast variance have the functional form required by Theorem 1; thus, the equal-proportional allocation optimizes $(P1)$ when using these objectives.

COROLLARY EC.1. *Assume all viewer types share the same mean number of impressions per arrival* ($\mu_v = \mu \; \forall v \in V$)*, and that as a function of the planned proportion* p*, the probability a given campaign is served to a given arrival is the same across all viewer types* ($\beta_v(p) = \beta(p) \; \forall v \in V$)*. Then if* $\beta(p)$ *is concave in* p*, the equal-proportion allocation* $\mathbf{p} = \mathbf{q}$ *is optimal for* $(P1)$ *under the objective of maximizing expected reach.*

Proof. From Equation (5), total expected reach is $f_0(\mathbf{p}) = \sum_{k\in K, v\in V_k} \frac{\lambda_v}{\eta}\left(1-e^{-\eta\beta(p_{vk})}\right)$. Because $\max f_0(\mathbf{p}) \equiv \min(-\mu\eta) f_0(\mathbf{p}) \equiv \min \sum_{k\in K, v\in V_k} \mu\lambda_v \left(e^{-\eta\beta(p_{vk})}-1\right) \equiv \min \sum_{k\in K, v\in V_k} s_v \left(e^{-\eta\beta(p_{vk})}-1\right)$, the result follows from Theorem 1 using $h(p) \stackrel{\text{def}}{=} e^{-\eta\beta(p)}-1$. Note that $\beta(p)$ concave $\implies -\eta\beta(p)$ convex $\implies e^{-\eta\beta(p)}$ convex $\implies h(p)$ convex. $\square$

COROLLARY EC.2. *Assume all viewer types share the same mean and standard deviation for the number of impressions per arrival ($\mu_v = \mu$, $\sigma_v = \sigma\ \forall v \in V$), and as a function of the planned proportion p, the second moment of the random fraction of impressions awarded to a given campaign is the same across all viewer types ($\alpha_v(p) = \alpha(p)\ \forall v \in V$). Then if $\alpha(p)$ is convex in p, the equal-proportion allocation $\mathbf{p} = \mathbf{q}$ is optimal for (P1) under the objective of minimizing stochastic variance.*

Proof. From Equation (14), total stochastic variance is $f_0(\mathbf{p}) = \sum_{k\in K, v\in V_k} (\sigma^2+\mu^2)\lambda_v \alpha(p_{vk})$. Since $\min f_0(\mathbf{p}) \equiv \min \frac{\mu}{\sigma^2+\mu^2} f_0(\mathbf{p}) \equiv \min \sum_{k\in K, v\in V_k} \mu\lambda_v \alpha(p_{vk}) \equiv \min \sum_{k\in K, v\in V_k} s_v \alpha(p_{vk})$, the result follows from Theorem 1 using $h(p) \stackrel{\text{def}}{=} \alpha(p)$. $\square$

COROLLARY EC.3. *Assume all viewer types share the same mean number of impressions per arrival ($\mu_v = \mu\ \forall v \in V$). Then if $\text{Var}[\Lambda_v] = \theta\lambda_v\ \forall v \in V$, where $\theta \geq 0$ is a constant, the equal-proportion allocation $\mathbf{p} = \mathbf{q}$ is optimal for (P1) under the objective of minimizing forecast variance.*

Proof. From Equation (15), total forecast variance is $f_0(\mathbf{p}) = \sum_{k\in K, v\in V_k} \mu^2 p_{vk}^2 \text{Var}[\Lambda_v]$. Since $\min f_0(\mathbf{p}) \equiv \min \frac{1}{\mu} f_0(\mathbf{p}) \equiv \min \sum_{k\in K, v\in V_k} \mu\theta\lambda_v p_{vk}^2 \equiv \min \sum_{k\in K, v\in V_k} s_v p_{vk}^2$, the result follows from Theorem 1 using $h(p) \stackrel{\text{def}}{=} p^2$. $\square$

EC.4. Quadratic Transportation Problem Duality

Consider the following transportation problem with quadratic objective. Each source $s \in S$ is connected to the set of sinks $t \in T_s$; likewise, each sink $t \in T$ is connected to the set of sources $s \in S_t$. The cost of transporting x_{st} units from source s to sink t is $c_{st}x_{st}^2$, where we assume $c_{st} > 0$. Source s can supply up to a_s units, and sink t demands exactly b_t units. The amount of flow on arc x_{st} is limited to the upper bound of d_{st}. This transportation problem is written:

$$\begin{aligned}
\min\ & \tfrac{1}{2}\textstyle\sum_{s\in S,t\in T_s} c_{st}x_{st}^2 \\
\text{s.t.}\ & \textstyle\sum_{t\in T_s} x_{st} \le a_s\ \forall s\in S \\
& \textstyle\sum_{s\in S_t} x_{st} = b_t\ \forall t\in T \\
& 0\le x_{st}\le d_{st}\ \forall s\in S, t\in T_s
\end{aligned}$$

In standard form, the problem is:

$$\begin{aligned}
\min\ & \tfrac{1}{2}\textstyle\sum_{s\in S,t\in T_s} c_{st}x_{st}^2 && \underline{\text{Dual Vars}} \\
\text{s.t.}\ & \textstyle\sum_{t\in T_s} x_{st} - a_s \le 0\ \forall s\in S && \ldots u_s \\
& b_t - \textstyle\sum_{s\in S_t} x_{st} = 0\ \forall t\in T && \ldots v_t \\
& -x_{st}\le 0\ \forall s\in S, t\in T_s && \ldots w_{st} \\
& x_{st} - d_{st}\le 0\ \forall s\in S, t\in T_s && \ldots z_{st}
\end{aligned}$$

The Lagrangian is therefore:

$$\begin{aligned}
L(x,u,v,w,z) &= \frac{1}{2}\sum_{s\in S,t\in T_s} c_{st}x_{st}^2 + \sum_{s\in S}\left(\sum_{t\in T_s} x_{st} - a_s\right)u_s + \sum_{t\in T}\left(b_t - \sum_{s\in S_t} x_{st}\right)v_t \\
&\quad + \sum_{s\in S,t\in T_s}(-x_{st})\,w_{st} + \sum_{s\in S,t\in T_s}(x_{st}-d_{st})\,z_{st} \\
&= L_0 + \sum_{s\in S,t\in T_s} L_{st}(x_{st}),\ \text{where}
\end{aligned}$$

$$L_0 = -\sum_{s\in S} a_s u_s + \sum_{t\in T} b_t v_t - \sum_{s\in S,t\in T_s} d_{st}z_{st}; \qquad L_{st}(x_{st}) = \frac{1}{2}c_{st}x_{st}^2 + (u_s - v_t - w_{st} + z_{st})x_{st}.$$

Since $\partial L/\partial x_{st} = \partial L_{st}(x_{st})/\partial x_{st} = c_{st}x_{st} + u_s - v_t - w_{st} + z_{st}$, the Karush-Kuhn-Tucker conditions are:

$$\begin{aligned}
\text{(Stationarity)}\ & c_{st}x_{st} = -u_s + v_t + w_{st} - z_{st}\ \forall s\in S, t\in T_s \\
\text{(Primal feasibility)}\ & \textstyle\sum_{t\in T_s} x_{st}\le a_s\ \forall s\in S \\
& \textstyle\sum_{s\in S_t} x_{st} = b_t\ \forall t\in T \\
& 0\le x_{st}\le d_{st}\ \forall s\in S, t\in T_s \\
\text{(Dual feasibility)}\ & u_s\ge 0\ \forall s\in S \\
& w_{st}\ge 0\ \forall s\in S, t\in T_s \\
& z_{st}\ge 0\ \forall s\in S, t\in T_s \\
\text{(Complimentary slackness)}\ & \left(\textstyle\sum_{t\in T_s} x_{st} - a_s\right)u_s = 0\ \forall s\in S \\
& x_{st}w_{st} = 0\ \forall s\in S, t\in T_s \\
& (x_{st}-d_{st})z_{st} = 0\ \forall s\in S, t\in T_s
\end{aligned}$$

We now solve for the dual objective. First, we note the Lagrangian dual function is $g(u,v,w,z) := \inf_x L(x,u,v,w,z) = L_0 + \sum_{s\in S,t\in T_s}\inf_{x_{st}} L_{st}(x_{st})$. Since L_{st} is convex in x_{st}, a minimum is obtained by solving the first order condition $\partial L_{st}/\partial x_{st} = 0$. Since $c_{st} > 0$, the first order condition yields

$x^*_{st} = \frac{1}{c_{st}}(-u_s + v_t + w_{st} - z_{st})$. Therefore, at optimality, $L_{st}(x^*_{st}) = -\frac{1}{2c_{st}}(-u_s + v_t + w_{st} - z_{st})^2$. Hence, the Lagrangian dual function is:

$$g(u,v,w,z) = -\frac{1}{2}\sum_{s\in S, t\in T_s}\frac{1}{c_{st}}(-u_s + v_t + w_{st} - z_{st})^2 - \sum_{s\in S} a_s u_s + \sum_{t\in T} b_t v_t - \sum_{s\in S, t\in T_s} d_{st} z_{st}.$$

The dual problem is therefore:

$$\begin{aligned}
\max\ & -\tfrac{1}{2}\textstyle\sum_{s\in S,t\in T_s}\frac{1}{c_{st}}\left(-u_s + v_t + w_{st} - z_{st}\right)^2 - \sum_{s\in S} a_s u_s + \sum_{t\in T} b_t v_t - \sum_{s\in S,t\in T_s} d_{st} z_{st} \\
\text{s.t. } & u_s \geq 0\ \forall s \in S \\
& w_{st} \geq 0\ \forall s \in S, t \in T_s \\
& z_{st} \geq 0\ \forall s \in S, t \in T_s
\end{aligned}$$

EC.5. GTDA Aggregation Theory Proofs

EC.5.1. Proof of Theorem 2

THEOREM 2. $z^{AUX(1/\mathbf{s})} \leq z^{OPP}$.

Proof. Using η, γ, θ, and ϕ as the Lagrange multipliers for the impression goal constraints, supply constraints, variable lower bounds (nonnegativity), and variable upper bounds, respectively, the Lagrangian dual for $(AUX(1/\mathbf{s}))$ is (see §EC.4 for the full derivation):

$$\begin{aligned}
(DAUX(1/\mathbf{s}))\ \max\ & -\frac{1}{2}\sum_{\substack{k\in K,\\ i\in I_k}} s_{ik}\left(-\gamma_i + \eta_k + \theta_{ik} - \phi_{ik}\right)^2 - \sum_{i\in I} s_i\gamma_i + \sum_{k\in K} g_k\eta_k - \sum_{\substack{k\in K,\\ i\in I_k}} s_{ik}\phi_{ik} \\
& \text{s.t. } \gamma_i \geq 0\ \forall i \in I, \text{ and } \theta_{ik} \geq 0,\ \phi_{ik} \geq 0\ \forall k \in K, i \in I_k
\end{aligned}$$

Similarly, the dual of (OPP^{IMP}) with redundant variable upper bounds $x_{vk} \leq s_v\ \forall k \in K, v \in V_k$ is:

$$\begin{aligned}
(DOPP^{IMP})\ \max\ & -\frac{1}{2}\sum_{\substack{k\in K,\\ v\in V_k}} s_v\left(-\gamma_v + \eta_k + \theta_{vk} - \phi_{vk}\right)^2 - \sum_{v\in V} s_v\gamma_v + \sum_{k\in K} g_k\eta_k - \sum_{\substack{k\in K,\\ v\in V_k}} s_v\phi_{vk} \\
& \text{s.t. } \gamma_v \geq 0\ \forall v \in V, \text{ and } \theta_{vk} \geq 0,\ \phi_{vk} \geq 0\ \forall k \in K, v \in V_k
\end{aligned}$$

Consider an optimal solution $(\eta^*, \gamma^*, \theta^*, \phi^*)$ of $(DAUX(1/\mathbf{s}))$ which has value $z^{AUX(1/\mathbf{s})}$. Let $\widehat{\eta}_k = \eta^*_k\ \forall k \in K$, $\widehat{\gamma}_v = \gamma^*_{i(v)}\ \forall v \in V$, $\widehat{\theta}_{vk} = \theta^*_{i(v),k}\ \forall k \in K, v \in V_k$, and $\widehat{\phi}_{vk} = \phi^*_{i(v),k}\ \forall k \in K, v \in V_k$. Clearly $(\widehat{\eta}, \widehat{\gamma}, \widehat{\theta}, \widehat{\phi})$ is feasible for $(DOPP^{IMP})$. Evaluating $(\widehat{\eta}, \widehat{\gamma}, \widehat{\theta}, \widehat{\phi})$ in the objective function of $(DOPP^{IMP})$, we have:

$$-\frac{1}{2}\sum_{\substack{k\in K,\\ v\in V_k}} s_v\left(-\widehat{\gamma}_v + \widehat{\eta}_k + \widehat{\theta}_{vk} - \widehat{\phi}_{vk}\right)^2 - \sum_{v\in V} s_v\widehat{\gamma}_v + \sum_{k\in K} g_k\widehat{\eta}_k - \sum_{\substack{k\in K,\\ v\in V_k}} s_v\widehat{\phi}_{vk}$$

$$
\begin{aligned}
&= -\frac{1}{2}\sum_{\substack{k\in K,\\ i\in I_k}}\sum_{v\in V_{ik}} s_v\left(-\gamma^*_{i(v)}+\eta^*_k+\theta^*_{i(v),k}-\phi^*_{i(v),k}\right)^2-\sum_{i\in I}\sum_{v\in V_i} s_v\gamma^*_{i(v)}+\sum_{k\in K} g_k\eta^*_k-\sum_{\substack{k\in K,\\ i\in I_k}}\sum_{v\in V_{ik}} s_v\phi^*_{i(v),k}\\
&= -\frac{1}{2}\sum_{\substack{k\in K,\\ i\in I_k}}\left(-\gamma^*_i+\eta^*_k+\theta^*_{ik}-\phi^*_{ik}\right)^2\sum_{v\in V_{ik}} s_v-\sum_{i\in I}\gamma^*_i\sum_{v\in V_i} s_v+\sum_{k\in K} g_k\eta^*_k-\sum_{\substack{k\in K,\\ i\in I_k}}\phi^*_{ik}\sum_{v\in V_{ik}} s_v\\
&= -\frac{1}{2}\sum_{\substack{k\in K,\\ i\in I_k}} s_{ik}\left(-\gamma^*_i+\eta^*_k+\theta^*_{ik}-\phi^*_{ik}\right)^2-\sum_{i\in I} s_i\gamma^*_i+\sum_{k\in K} g_k\eta^*_k-\sum_{\substack{k\in K,\\ i\in I_k}} s_{ik}\phi^*_{ik}\\
&= z^{AUX(1/\mathbf{s})}
\end{aligned}
$$

Hence, there exists a dual feasible point with objective value $z^{AUX(1/\mathbf{s})}$ in $(DOPP^{IMP})$. Since $(DOPP^{IMP})$ is a maximization problem, $z^{OPP}\geq z^{AUX(1/\mathbf{s})}$. $\square$

EC.5.2. Proof of Theorem 3

THEOREM 3. *The solution* $\mathbf{p}^j$ *returned by* REFINEPARTITIONANDSOLVE *is optimal in* (OPP).

Proof. Say REFINEPARTITIONANDSOLVE terminates at iteration j with partition I^j. Furthermore, let $\mathbf{x}^j$ and $z^{AUX(1/\mathbf{s})}$ be an optimal solution and corresponding optimal value for $(AUX(1/\mathbf{s}))$ under partition I^j, as computed in iteration j. We now show that $(\mathbf{x}^j,\mathbf{c})=(\mathbf{x}^j,1/\mathbf{s})$ is feasible for problem (APP^{IMP}) with partition I^j. First, it is clear that since $(AUX(1/\mathbf{s}))$ has impression goal and nonnegativity constraints, those constraints in (APP^{IMP}) are satisfied by $\mathbf{x}^j$. We just need to verify that $(\mathbf{x}^j,1/\mathbf{s})$ satisfies the nonlinear yield constraints Y^{IMP} of (APP^{IMP}). But since REFINEPARTITIONANDSOLVE always terminates with no viewer types overallocated (i.e. $\overline{y}^j_v=1\,\forall v\in V$), we know that $(\mathbf{x}^j,1/\mathbf{s})\in Y^{IMP}$: to verify, substitute $y_v=1\,\forall v\in V$, $a_{ik}=b_{ik}=s_{ik}\,\forall k\in K, i\in I_k$, and $p_{ik}=p^j_{ik}=x^j_{ik}/s_{ik}\,\forall k\in K, i\in I_k$ into Y^{IMP}. Thus, $(\mathbf{x}^j,1/\mathbf{s})$ is feasible for problem (APP^{IMP}) with partition I^j.

Now we evaluate $(\mathbf{x}^j,1/\mathbf{s})$ in the objective of (APP^{IMP}). Since $c_{ik}=1/s_{ik}\,\forall k\in K, i\in I_k$, the objectives of $(AUX(1/\mathbf{s}))$ and (APP^{IMP}) are identical. Hence, not only is $(\mathbf{x}^j,1/\mathbf{s})$ feasible in (APP^{IMP}), but this solution has value $z^{AUX(1/\mathbf{s})}$. Therefore, $z^{APP}\leq z^{AUX(1/\mathbf{s})}$. But from Proposition 2 and Theorem 2 we know that for the fixed inventory block partition I^j, $z^{AUX(1/\mathbf{s})}\leq z^{OPP}\leq z^{APP}$. Hence, $z^{AUX(1/\mathbf{s})}=z^{OPP}=z^{APP}$.

Therefore, we have shown that the disaggregation of $\mathbf{x}^j$ is optimal in (OPP^{IMP}); correspondingly, the disaggregation of $\mathbf{p}^j$ is optimal in (OPP^{PROP}). $\square$

EC.6. Additional Computational Results

This section reports the measured performance of RefinePartitionAndSolve and GetCloseAndScaleUp on the six test cases from Figure 5. The details are organized in Figure EC.1. For example, in the "Large, Locally Tight" case, 10 refinement iterations produces a solution that is on average 99.9%-feasibile, requiring an average of 991 inventory blocks and 26 seconds (row C); 6 refinement iterations are required before the scaling algorithm achieves 100% success, and this requires 64 inventory blocks and 9 seconds on average (row D); to get within 0.1% of proven optimality in 90% of instances, 8 refinement iterations were needed, which on average generated 254 inventory blocks in 14 seconds and subsequently performed 18 scaling steps in 31 seconds (row F); finally, to get within 0.1% of measured optimality in 90% of instances, 8 refinement steps were also required (row H).

	Small			Large		
	Loose	Globally Tight	Locally Tight	Loose	Globally Tight	Locally Tight
(A)	(0, 1, 1)	(0, 1, 1)	(2, 4, 1)	(0, 1, 14)	(0, 1, 14)	(2, 4, 4)
(B)	(0, 1, 1)	(3, 8, 4)	(6, 64, 1)	(0, 1, 14)	(3, 8, 54)	(6, 64, 9)
(C)	(2, 3, 3)	(7, 122, 10)	(10, 980, 8)	(2, 3, 42)	(7, 123, 108)	(10, 991, 26)
(D)	(0, 1, 1)	(0, 1, 1)	(7, 128, 2)	(0, 1, 14)	(0, 1, 14)	(6, 64, 9)
(E)	(0, 1, 1), (2, 2)	(2, 4, 3), (7, 7)	(7, 128, 2), (23, 3)	(0, 1, 14), (2, 27)	(2, 4, 40), (7, 89)	(6, 64, 9), (52, 63)
(F)	(0, 1, 1), (2, 2)	(5, 32, 7), (4, 4)	(8, 253, 3), (16, 3)	(0, 1, 14), (2, 27)	(5, 32, 80), (4, 55)	(8, 254, 14), (18, 31)
(G)	(0, 1, 1), (2, 2)	(0, 1, 1), (16, 16)	(7, 128, 2), (23, 3)	(0, 1, 14), (2, 27)	(0, 1, 14), (16, 204)	(6, 64, 9), (52, 63)
(H)	(0, 1, 1), (2, 2)	(5, 32, 7), (4, 4)	(8, 253, 3), (16, 3)	(0, 1, 14), (2, 27)	(5, 32, 80), (4, 55)	(8, 254, 14), (18, 31)

Figure EC.1 Summary of algorithm performance. (A-C) = # of Refine (Iterations, Blocks, Seconds) to achieve (A) 90%, (B) 99%, and (C) 99.9% feasibility; (D) = # of Refine (Iterations, Blocks, Seconds) to achieve 100% scaling success; (E-F) = # of Refine (Iterations, Blocks, Seconds) and # of Scaling (Iterations, Seconds) to get within (E) 1%, (F) 0.1% of proven optimality; (G-H) = # of Refine (Iterations, Blocks, Seconds) and # of Scaling (Iterations, Seconds) to get within (G) 1%, (H) 0.1% of measured optimality.