PREPRINT

Impression Allocation under Campaign Quotas: A Persistence Rule for Segmentation and a Regime Map for Pacing

Alexander Apartsin[1] · Yehudit Aperstein[2]

[1]School of Computer Science, Faculty of Sciences, Holon Institute of Technology (HIT), Holon, Israel

[2]Intelligent Systems, Afeka Academic College of Engineering, Tel-Aviv, Israel

ABSTRACT

On platforms that own rich first-party audience data, a mobile operator or a retail-media network, advertising is sold as guaranteed delivery: each campaign books a fixed number of impressions to a finely targeted audience, and the platform must fill that quota. A single visitor belongs to many of these audiences at once, so several campaigns are eligible for the same impression while only one can be served, and the most valuable audiences are the most contested. Serving each impression to whichever campaign is most likely to click it now spends these scarce, high-response segments on broadly popular campaigns, so campaigns that do well only on those segments are left with a poor match; the loss is a failure to coordinate, not a failure to predict. With the quotas fixed, what makes one allocation better than another is not how campaigns or audiences perform on average, but how well specific campaign-and-audience pairs fit, and only the part of that fit that stays stable from one time period to the next. This yields a rule a platform can commit to in advance: build segments from the pairings whose fit is stable, at the resolution the data can estimate, then plan the allocation and re-solve it as delivery proceeds.

On three public advertising logs (Taobao, Avazu, Outbrain), with a replay method guarded against grading a plan on its own estimates and verified against a synthetic market whose answer is known, planning this way beats both greedy serving and learning delivery prices online, which fails under heavy demand. One condition sets the payoff: audiences must be contested; where

supply is ample, planning adds little. The recipe is three steps: choose segments by stable fit, measure contention, and obtain prices by solving and re-solving the plan.

1 Introduction

A mobile advertising platform must answer one question millions of times a day: *a subscriber has just opened a page; which campaign should fill this impression?* The obvious answer, show whichever ad this subscriber is most likely to click, is wrong. Campaigns carry contracted impression quotas, subscribers arrive in an unpredictable order, and every impression given to one campaign is an impression denied to all others. Serving each impression greedily fills popular campaigns with the first subscribers who arrive, not those who respond best, and leaves the remaining campaigns with a poorly matched audience.

On the open web this failure is blunted by coarseness: broad audiences are interchangeable audiences, so the opportunity cost of a greedy choice is small. Operator-owned advertising markets invert that condition. A platform that owns logged-in, first-party identity, a mobile operator, or, in its largest contemporary form, a retail-media network selling onsite display against its own commerce data, supports far finer segmentation, and fine segmentation is precisely what advertisers pay it for: not “males 18–49” but heavy category buyers on related content in the promotional window. Its campaigns are sold direct as guaranteed impression quotas with makegood penalties, and it controls the whole serving stack, with no auction standing between the plan and the impression. This study applies to that class of market. Its originating instance is the mobile-operator system of Section 6, its largest current instance is retail media, and the public commerce-media log we evaluate on (Taobao) comes from inside it.

Precision is what turns serving into an allocation problem. Narrow, high-value targets are scarce, and campaigns defined this precisely overlap: the subscribers one advertiser most wants are the subscribers other advertisers most want, in the same hours, on the same content. Contention over specific sub-audiences stops being an edge case and becomes the normal operating condition. Every impression then carries a real opportunity cost, and greedy serving, harmless when audiences are interchangeable, systematically hands the scarcest inventory to whichever campaign arrives first. The same richness that creates the problem also makes it solvable: with complete attribution, the click probability of every (campaign, segment) pair is a measured quantity, so the cost of any assignment against any alternative can be computed rather than guessed. An operator that targets precisely and attributes completely does not merely benefit from

global optimization; it forfeits the value of its own data without it. Precision here is a property of *demand*: advertisers buy narrow targets, and narrow targets contend. Which segmentation the allocator itself should use is a separate, empirical question, and Section 4 shows its answer is governed by which interactions persist, not by maximal fineness.

This paper treats ad serving as that **global allocation problem** and makes three contributions, each validated on public advertising logs. First, a pre-committable segmentation rule: measure the persistence of campaign-by-segment interaction residuals across adjacent planning windows, segment only on the axes that persist, and scale granularity to per-cell data density. Second, a replay evaluation protocol for allocation policies, hardened against the winner's curse and selection artifacts that we show inflate naive replay. Third, a regime map locating where planning pays: plan-informed policies lead at exact feasibility and dominate outright under over-subscription, where online dual-price learning collapses and solving for the prices does not. The map holds on a second dataset and under a revenue objective with delivery guarantees.

Together these compose a three-step rule that the planning team of any retail-media network or operator-owned platform can act on directly. **Probe persistence** to choose the segmentation: keep the axes whose campaign-by-segment interaction persists across planning windows, at the depth per-cell data supports. **Measure contention** to decide whether allocating is worth the effort at all: compare each segment's supply to the quotas booked against it, because an uncontended market yields little to allocate whatever the segmentation. **Solve for the prices** when it is contended: obtain the dual prices by solving the plan and re-solving on the pacing cadence, rather than learning them online, which is the only price source robust across every contention level, dataset, and objective we test. The rest of the paper establishes each step and the evidence behind it.

Section 2 states the allocation program and its interaction-invariance structure. Section 3 develops the replay protocol and validates it against a known ground truth, Section 4 the persistence rule, and Section 5 the regime map with its cross-dataset and revenue-objective replications. Section 6 summarizes the reference serving architecture that motivated the study, Section 7 situates the work in the online-allocation and offline-evaluation literature, and Sections 8 and 9 close with design principles and scope. The appendix collects the worked example (A.1), the proof of Proposition 1 (A.2), the simulation calibration (A.3), the extended robustness checks (A.4), and the full serving architecture (A.5).

2 Formulation

Formally, the planning stage solves a transportation-style optimization. Audience supply is described by segments: s ranges over (cluster, context slot) pairs, where a cluster is a group of behaviorally similar subscribers and a context slot is a recurring situation (a time window crossed with a content category). Let H_s be the predicted number of impressions segment s will generate during the planning horizon, and $p_{c,s}$ the estimated probability that an impression from segment s produces a click on campaign c. Each campaign pays r_c per impression and q_c per click, and has bought Q_c impressions. The plan chooses $x_{c,s}$, the number of impressions of campaign c allocated to segment s:

$$
\begin{aligned}
\text{maximize} \quad & \Sigma_{c,s} \; x_{c,s} \cdot (r_c + q_c \cdot p_{c,s}) \\
\\
\text{subject to} \quad & \Sigma_c \; x_{c,s} \;\le\; H_s \qquad \text{(segment supply)} \\
& \Sigma_s \; x_{c,s} \;\le\; Q_c \qquad \text{(campaign quota)} \\
& x_{c,s} = 0 \text{ where targeting excludes } (c, s) \\
& x_{c,s} \;\ge\; 0
\end{aligned}
$$

Figure 2 is this program with two campaigns, four single-subscriber segments, $H_s = 1$ and $Q_c = 2$. At production scale the same structure holds; the segments are just coarser and the counts larger. Because both objective and constraints are linear, the plan is solvable with off-the-shelf LP solvers (we use the dual simplex of [1]) even for large instances, and, critically, it is solved **offline**. The online server never optimizes; it executes a pre-computed plan and corrects for drift.

Three practical notes. First, the objective prices impressions and clicks separately, so campaigns that pay mostly per impression and campaigns that pay mostly per click compete in one currency, expected revenue; when supply suffices, quotas bind at the optimum (every unit of x earns at least $r_c > 0$), and the impression-price term becomes a constant, making the allocation choice a pure expected-clicks problem; Section 5 studies the over-subscribed regime where this simplification fails. Second, the quota appears as an upper bound: when targeting and supply cannot fill a quota, the program returns the revenue-maximal under-delivery rather than infeasibility, and the shortfall is the input to make-good scheduling, which we treat as outside the model. Third, the quota constraint is what makes greedy serving inadequate: without it, serving every impression to its highest-value campaign would be optimal, and no planning stage would be needed.

2.1 Interaction invariance under fixed margins

The value matrix admits an additive decomposition into a campaign main effect, a segment main effect, and an interaction residual,

$$v_{cs} = a_c + b_s + g_{cs},$$

where g is **double-centered**: its supply-weighted average over segments is zero for every campaign, and over campaigns is zero for every segment. This decomposition is unique and always exists; a_c is campaign c's average value, b_s segment s's deviation, and g_{cs} the part of the value specific to the pairing. The following fact is the paper's conceptual backbone.

Proposition 1 (interaction invariance). Let x and x' be any two feasible allocations with identical campaign margins ($\Sigma_s x_{c,s} = \Sigma_s x'_{c,s}$ for every c) and identical segment margins ($\Sigma_c x_{c,s} = \Sigma_c x'_{c,s}$ for every s). Then their values differ only through the interaction term:

$$V(x) - V(x') = \Sigma_{c,s} \left(x_{c,s} - x'_{c,s} \right) \cdot g_{c,s}.$$

The proof is a two-line expansion of the objective and appears in Appendix A.5.

The consequence is that **representation choice for allocation should target the stable interaction component, not marginal click-rate predictiveness**: a segmentation that sharpens the estimates of a_c and b_s but not of g_{cs} cannot change a fixed-margin assignment, however much it improves overall CTR prediction. Section 4 turns this into a selection rule. Four corollaries fix the scope precisely.

1. **Click maximization.** With $v_{cs} = p_{cs}$, the entire allocation value under fixed margins is carried by the interaction of click probability; campaign and segment average click rates are decision-irrelevant.
2. **Fixed per-impression contract value.** When campaign c pays a fixed r_c per delivered impression, that revenue contributes $\Sigma_c r_c Q_c$, a constant under fixed campaign margins, and the allocation reduces to the expected-clicks problem, as noted above.
3. **Slack margins.** Segment margins are fixed by the arriving stream (supply is exogenous and identical across policies), so the segment main effect always cancels. If campaign quotas do not

all bind, campaign totals differ between allocations and the campaign main effect re-enters, bounded by $|V(x)-V(x') - \text{interaction term}| \le \|a\|_\infty \cdot \|\Delta Q\|_1$, where ΔQ is the difference in delivered campaign totals. Interaction invariance is therefore exact under full delivery and degrades gracefully with the delivery gap.

4. **Over-subscription.** When quotas exceed deliverable supply, campaign totals become endogenous, the bound above is loose, and a policy must decide which campaigns to under-deliver; the full objective (including r_c and any under-delivery penalty) governs, and interaction invariance no longer applies. This is exactly the regime the contention experiments of Section 5 probe, and it is why policy orderings there differ from the feasible case.

3 Replay evaluation protocol

A synthetic market controls everything (Appendix A.3); real logs control nothing, which is the point. The core experiments replay the public Taobao display-advertising dataset [2, 3]: 26.6 million impressions over 8 days in May 2017, roughly 1.1 million users with declared demographics, and explicit campaign identifiers, an in-vertical commerce-media log rather than a distant analog to an operator's data (Sections 4 and 5 add Avazu and Outbrain as the churn and contention boundaries). The market is reconstructed from the log itself: campaigns are the top 100 by planning-window volume, each campaign's quota is the number of impressions it actually received in the serving window, and policies re-pair the identical impression stream, so the counterfactual is *the same inventory, re-allocated*.

Replay evaluation has a trap. If the table of click probabilities that the planner optimizes also scores the outcome, both the plan and the bound harvest the table's estimation noise: in our first run the measured lift was +8.9% while the mean assignment value exceeded the best campaign's true click rate, an impossibility. The design that survives this is **dual-model scoring**: policies plan and serve using only planning-window estimates, while all values, including the hindsight bound, are scored under an independent table fitted on the serving window. A uniform-probability control run then yields exactly 0.0% lift with every policy at exactly 100% of the bound. This is a mechanical identity, not an empirical finding: by Proposition 1, a scorer constant across segments carries no interaction, so under fixed quotas every feasible assignment scores equally, and the control confirms the harness contains no implementation path that manufactures lift from a flat table. Evaluation is **walk-forward**: for each of five serving days, the plan is fitted on all prior days and scored on that day alone. The adaptive baseline is fully specified: a dual-price pacing policy

that serves the quota-remaining campaign maximizing $\hat{p}_{cs} - \mu_c$, updating $\mu_c \leftarrow \mu_c + \eta(\text{served}_c/Q_c - t/T)$ after each impression with η set to half the global click rate; it consults no plan. Reported errors are the scoring table's closed-form binomial variance propagated through each comparison's assignment difference; the across-fold spread of the five per-fold lifts independently reproduces the same standard error, and the all-folds-positive sign pattern supports the parametric statement distribution-free. The uniform control defends the harness; a further check defends the scorer itself: standardizing the scoring table along the strongest observable within-cell covariate the segmentation does not use (each user's own training-window click propensity) shifts the headline lifts by at most +0.3pp, upward, so observable within-cell selection in the logged exposures does not account for the measured gains. Support is also measured directly: only 6% of the planned policy's assignment mass (and 5% of the re-paired mass) lands in cells with fewer than 100 logged serving-window impressions, and scoring those thin cells with a pooled campaign-level fallback moves the headline from +3.46% to +2.99%; the counterfactual rests almost entirely on well-supported cells. (Forcing the fallback on all cells below 1,000 impressions, 78% of the mass, collapses the lift toward zero, as it must: a campaign-constant scorer deletes the interaction dimension, and under fixed quotas every assignment then scores equally, the same argument as the uniform control.) The replay isolates the architecture's allocation core, plan against serving policy at segment granularity: every campaign is eligible on every segment, and the per-subscriber serving components of Appendix A.5 (soft-membership blending, frequency caps, threshold pacing) operate below this granularity and are not exercised by these experiments.

3.1Does model-scored replay recover the truth?

Real logs cannot say whether the model-scored lifts equal the counterfactual truth, because the truth is unobserved. A semi-synthetic benchmark can. We build a world where the response is known: on a slice disjoint from all reported experiments (Taobao days 6–8) we fit a shrunk (campaign, segment) click surface, amplify its double-centered interaction by a factor κ so policies genuinely differ, and freeze the result as a ground-truth θ_{cs}. The logging policy is the real exposure pattern; clicks are then sampled $\sim$ Bernoulli(θ) on the real exposures to fit a planning table (days 9–11) and an independent scoring table (days 12–13), so both tables carry realistic estimation noise and θ itself never scores anything the planner sees. Five policies (greedy, static plan, periodic re-solve, fixed optimal dual, entropic Dual Mirror Descent) then replay the day 12–13 stream, and each assignment is scored twice: under the noisy table, exactly as the paper reports, and under θ, which only this controlled setting makes available. At realistic interaction strength

the model-scored replay recovers the decision that matters: every strong policy is correctly separated from greedy, the model-scored ranking matches the true ranking (Spearman 1.00 at κ=1.0, 0.90 at κ=1.5), the model selects the truly best policy in both cases, and the estimated greedy-relative lift sits within about 1.4 percentage points of its true value. The limit is also visible: when the interaction is amplified beyond the level seen in the real data (κ=2.0), the top policies bunch within roughly 0.3 points of one another under the true response, and the noisy table reorders those near-ties (Spearman 0.70), selecting the entropic dual over the static plan when the plan is in fact marginally better. Model-scored replay is therefore trustworthy for the claims this paper makes, which separate a strong tier from greedy and identify a robust re-solve or plan policy, and it should not be read as resolving fine differences among near-tied top policies. The recovery does not depend on a benign logging policy. Repeating the benchmark with a deliberately adversarial logger, one that selects on the outcome itself so high-response cells are over-observed and low-response cells starve, the worst case for a scorer fit on logged data, still identifies the true best policy at both signal strengths (rank correlation 0.90 and 1.00), at the cost of a small upward bias, under 1.2 percentage points, from the optimistic tables that outcome-selected logging produces. The benchmark and its adversarial variant are released with the replication package.

4 The persistence rule

Proposition 1 identifies the campaign-by-segment *interaction* as the only exploitable object under fixed margins. Applying it across windows adds a requirement the proposition does not: the interaction is estimated on the planning window and used on the serving window, so it must also *persist* between them. Campaign main effects persist strongly (week-over-week correlation 0.81) but Proposition 1 neutralizes them under fixed quotas; segment main effects are fixed by supply. Measuring the persistence of double-centered interaction residuals per candidate axis therefore predicts, from planning-window data alone, which segmentations can pay. The estimator is explicit. For a candidate axis, take two adjacent planning windows (here 8-day and 2-day slices of the training period) and, in each, form the empirical click rate of every (campaign, group) cell with at least 300 impressions in one window and 150 in the other. Double-center each window's cell rates by subtracting the impression-weighted campaign mean and the impression-weighted group mean and adding back the grand mean, leaving the pure interaction residual. The axis's persistence is the Pearson correlation of these residuals across the two windows over the shared cells; empty or below-threshold cells are dropped, not imputed. An axis is retained when its

persistence exceeds its estimation-noise floor (the correlation a pair of independent binomial draws at the same cell counts would produce, near zero at these volumes). The measured persistences:

Segmentation axis	Interaction persistence (r)	Interaction size beyond noise
Activity tier (usage volume)	0.48–0.54	~1pp on a 5.5pp base
Gender × age	0.28	~0.5pp
Hour × position slot	0.08	~0

The persistence table fixes the evaluation configuration *before* any replay is scored: the activity axis persists most strongly, and its measured persistence improves from four tiers to eight (r = 0.48 to 0.54), so the pre-committed configuration for the 100-campaign market is eight activity tiers. Both headline comparisons are reported for that single configuration; the remaining rows are the ablation that tests the rule.

Segments	Count	Planned vs greedy	z	Planned vs adaptive pacing	z
Activity, 8 tiers (pre-committed)	8	**+3.46% ± 0.69pp**	+5.0	**+0.92% ± 0.47pp**	+2.0
Activity, 4 tiers	4	+2.51% ± 0.50pp	+5.0	+1.15% ± 0.31pp	+3.7
Tier × demographics	168	+1.93% ± 0.91pp	+2.1	+0.40%	+0.6
Demographics only	48	−0.20% ± 0.61pp	−0.3	+0.46%	+1.1

Per-fold lifts for the pre-committed configuration, so signs are visible rather than summarized: against greedy +5.2, +4.2, +2.6, +3.2, +2.2 (five of five positive; one-sided sign-test p = 0.03 independent of the parametric error model); against adaptive pacing +2.2, +1.7, +1.9, −0.5, −0.9 (three of five positive, consistent with its z = 2.0).

Three conclusions. First, the architecture's claim holds on real inventory under the model-scored replay protocol: on the pre-committed configuration the plan beats greedy serving by 3.5% expected clicks (positive in every fold) and beats an adaptive dual-price pacing policy in the family of online allocation methods [5], the strong online baseline, by 0.9% (z = 2.0; the four-tier ablation

row separates from pacing at z = 3.7), while serving from precomputed tables. Second, **persistence, not granularity, carries the value**: eight segments on the persistent axis beat 168 finer ones, because finer cells add estimation noise and re-pairing churn faster than they add exploitable signal; and the right granularity scales with per-cell data density (in a 500-campaign market, four tiers beat eight). The density component is not free of tension: eight tiers give the larger edge over greedy, yet four tiers separate more decisively from the adaptive baseline, so the depth that maximizes the gap depends on which competitor one measures against; we pre-commit to the greedy-referenced choice and report both. Third, the persistence measurement itself is the deployable design rule: before segmenting, correlate double-centered interaction residuals across adjacent planning windows, and segment only on axes that persist. The rule generalizes: on a second public log (Avazu, 40.4M mobile impressions [4]), the same measurement ranks page category as the most persistent axis (r = 0.88) and, in both datasets, time-of-day as the least, confirming that time belongs to the pacing loop, not to segmentation. Avazu's own replay is where the rule earns its boundary: a plan fitted on a full prior window and served for a whole day is negative on three of five days, because Avazu's ad portfolio and delivery mix turn over within a day. The persistence measurement implies a re-planning timescale, and re-planning at that timescale (every 50,000 impressions) turns the same market positive on all five days (+6.5%, +8.3%, +1.7%, +2.7%, +0.5% over greedy), confirming the mechanism and the stability condition Section 9 states. With that rule applied, the replayed policies capture 85–91% of the hindsight bound; the residual is the estimation noise and week-over-week drift that the pacing loop of Appendix A.5 exists to absorb.

The granularity-density frontier. Varying tier count against market size traces the frontier directly. In the top-100 market, 8 tiers deliver +3.46% over greedy (z = 5.0) while 16 tiers deliver +0.32% (z = 0.4). In the top-500 market, 4 tiers deliver +1.55% (z = 4.8), 8 tiers +0.94% (z = 2.2), and 16 tiers −0.43% (z = −0.9). Optimal granularity shrinks monotonically as per-cell density falls, so the rule is: persistence picks the axis, density picks the depth.

These two components combine into a single selection procedure that runs on planning-window data alone and returns a segmentation, or abstains, before any serving outcome is seen.

Algorithm 1 (persistence-guided segmentation selection). Inputs: candidate axes A with nested resolutions d, two adjacent planning windows, a support tolerance τ, and a persistence floor ρ_0 set to the estimation-noise null.

1. For each (A, d), form double-centered interaction residuals in each window over cells meeting the minimum-count thresholds; estimate persistence $\rho_{A,d}$ as their correlation, with a bootstrap lower confidence bound.
2. Solve the planning-window LP once to get the planned allocation mass, and compute, per (A, d), the fraction of that mass landing in cells whose serving-window support would meet the minimum count.
3. Call (A, d) **admissible** if at least $(1 - \tau)$ of the planned mass is supported and the persistence lower bound exceeds ρ_0.
4. Among admissible axes pick the one with the strongest persistence; within it pick the finest depth whose cells retain enough data to resolve the interaction,

$$d^* = \max \{ d : \operatorname{median}_{(c,s):\, x^{plan}_{cs}>0} N^{train}_{cs} \geq n_{min} \}, \quad n_{min} \approx \bar{p}(1-\bar{p}) /$$

where N^{train} is the planning-window count of a cell, $\bar{p}$ the base click rate, and σ_γ the interaction standard deviation; n_{min} is the count at which a per-cell estimate resolves the interaction, both quantities measured on planning data. 5. If no axis is admissible, **abstain** (serve the unsegmented market). Freeze the choice before the serving window.

6. Independently of the axis choice, measure **contention** on the planning window: for each segment compare the quota booked against it to its forecast supply, κ_s = ($\sum_c$ planned demand on s) / S_s. Where supply dwarfs demand ($\kappa_s \ll 1$ on the segments that carry the interaction) the achievable allocation gain is bounded and small whatever the segmentation, because greedy already leaves little on the table; the persistence probe then identifies the right axis but the market does not reward acting on it. This is a pre-deployment quantity, computed before any serving, and it is what separates the two markets in Section 5: contended Taobao rewards the segmentation, uncontended Outbrain does not until the demand regime is stressed.

The axis step is decisive and robust: run blind it selects activity on Taobao, page platform on Outbrain, and the axis ranking Avazu's table implies. The Outbrain choice was frozen in a public, dated commit before its replay was scored, the sense in which that application is pre-registered. The depth step ties to a measurable density. The median planning count of the cells the plan fills falls with both depth and market breadth (Taobao top-100: about 2,700, 1,400, and 670 at 4, 8, and

16 tiers; top-500: 1,100, 530, and 260), and with $\sigma_\gamma \approx$ 1pp on a 5% base giving n_{min} on the order of 10^3, the criterion keeps eight tiers at top-100 and four at top-500, matching the ablation optima above. The threshold is an order-of-magnitude reliability target rather than a constant validated out of sample, so we report depth selection as density-guided rather than fully closed; the axis and abstain decisions are the parts of Algorithm 1 that run without any such tuning.

5 The regime map

Over-subscription is the regime that motivates planning, and the market reconstruction supports stressing it directly. Quotas set to each campaign's realized delivery make the market exactly feasible; scaling all quotas above deliverable supply forces genuine contention, and policies choose which campaigns under-deliver. Six policies, alongside greedy, replay each regime on the pre-committed configuration: the static plan; a periodically re-solved plan (the same LP re-solved on remaining quotas and prorated remaining supply every 20,000 impressions); a scalar dual-price pacing controller; the fixed optimal dual; and dual mirror descent in Euclidean and entropic geometries. The dual-price policies and their tuning are specified below. Lift versus greedy, combined over the five walk-forward folds, with z in parentheses:

Quota scale	Static plan	Re-solved plan	Fixed optimal dual	DMD, entropic	DMD, Euclidean	Scalar pacing
1.0	+3.46% (5.0)	+4.27% (6.1)	+4.45% (6.4)	+4.54% (6.8)	+4.73% (7.1)	+3.86% (6.3)
1.1	+2.16% (3.1)	+3.49% (5.2)	+2.41% (3.8)	+3.62% (5.7)	+2.65% (4.3)	+0.73% (1.4)
1.25	+1.56% (2.2)	+3.39% (5.2)	+3.05% (4.9)	+3.06% (5.4)	−0.25% (−0.4)	−2.25% (−4.0)
1.5	+0.28% (0.4)	+2.22% (3.4)	+1.49% (2.4)	−0.52% (−0.9)	−4.76% (−7.2)	−6.88% (−11.2)

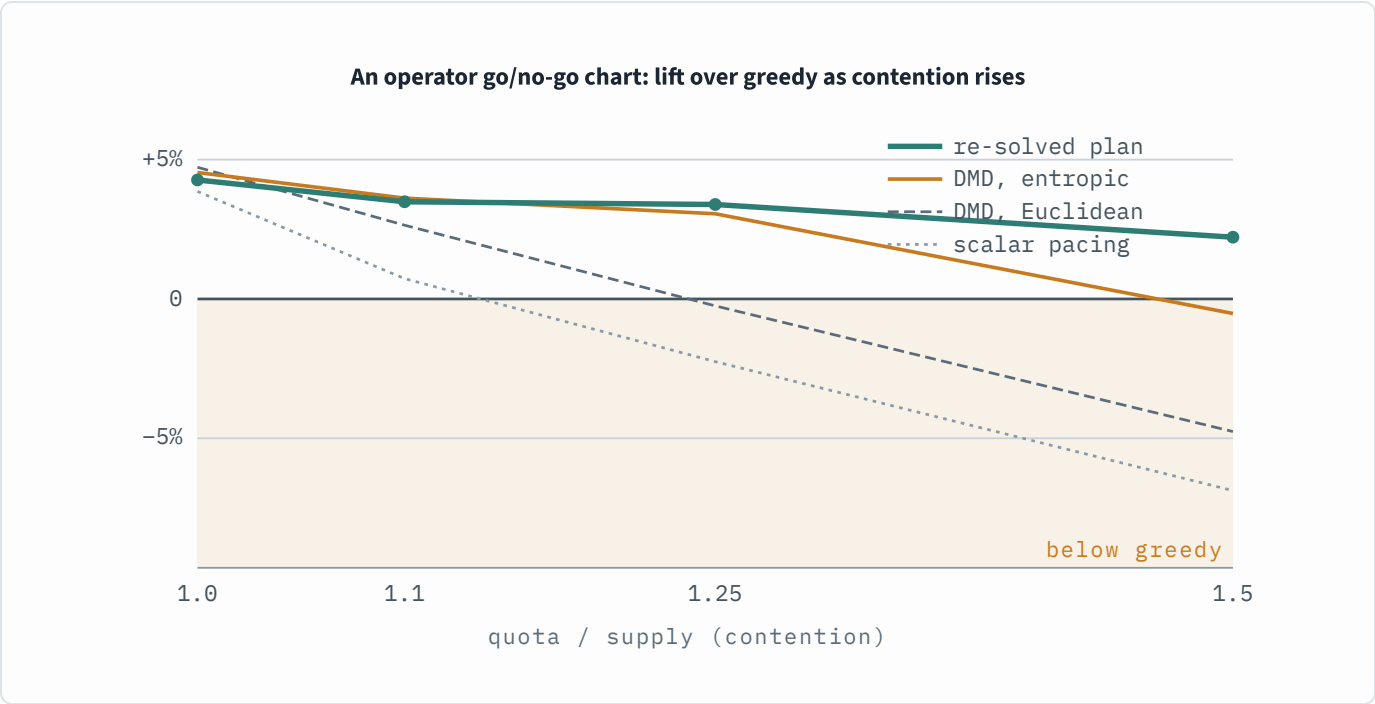

Figure 1. The regime map as a deployment chart. Each curve is one policy's lift over greedy serving (walk-forward mean, Taobao top-100, pre-committed 8-tier segmentation) as quotas are scaled above deliverable supply. Of the four policies plotted, the re-solved plan is the only one that stays above greedy across the whole contention range; every online price-learning method crosses into the shaded loss region, in order of update aggressiveness (scalar first, then Euclidean, then entropic mirror descent). The static plan and the fixed optimal dual, omitted to keep the chart legible, also stay non-negative under the click objective (Section 5 table). An operator reads the horizontal axis from the pre-deployment contention ratio of Algorithm 1, step 6, and the chart says when planning is worth the effort and which pacing to trust.

The dual-price baselines are given every advantage: their step size is tuned on held-out planning-window traffic, the warm variant is initialized from the plan LP's own quota shadow prices, and each paces toward a *feasible* delivery trajectory (the proportionally deliverable quota for the plan-free variants, the LP's planned delivery for the warm variant) rather than the raw over-subscribed quota. The re-solved plan is held to the same information as the plan: it re-solves on the planning-window forecast prorated to remaining supply, never on realized serving-window arrivals. Two further policies separate the price from its adaptation. The **fixed optimal-dual** policy serves the quota-remaining campaign maximizing $\hat{p}_{cs} - \mu^*_c$ using the exact optimal quota shadow prices μ^* of the over-subscribed plan LP, with no online update: a per-campaign-price policy that never adapts. Against it we run the state-of-the-art adaptive method, **dual mirror descent** [21] in both Euclidean and entropic geometries, tuned on the planning window; the earlier scalar quota-error controller is retained as a practical baseline.

At exact feasibility all seven policies cluster within a point; most of the value is the forecast itself, which each consumes. Under contention they separate, and the separation is a spectrum set by

how the prices are obtained (Figure 1). Adaptation degrades with the update rule: the scalar controller weakens first (+0.7% at 110% quotas, −6.9% at 150%), Euclidean mirror descent holds to 110% then falls (−4.8% at 150%), and even the strongest adaptive method, entropic mirror descent, tracks the re-solved plan through 125% over-subscription (+3.06% versus +3.39%) before turning slightly negative at 150% (−0.5%). The policies that obtain prices by solving rather than learning them online, the static plan, the re-solved plan, and the fixed optimal dual, are the ones that stay non-negative across the range; the re-solved plan holds the largest margin under contention (+2.2% to +4.3%), while the static plan's edge narrows to within noise at the heaviest over-subscription (+0.28% at 150%). The semi-synthetic benchmark of Section 3.1 bounds what this table can resolve: gaps at or below roughly 0.3 points, the feasibility row where every policy clusters and the re-solve-versus-entropic-DMD cell at 125%, are within the protocol's resolution and are not read here as an ordering; the collapses that carry the argument, 3 to 45 points, are far outside it. Two conclusions follow. First, the failure under heavy over-subscription is shared by every adaptive dual-price method we test, modern mirror descent included; a well-chosen adaptive method (entropic DMD) is nonetheless competitive through moderate contention, so the claim is about the heavy-contention regime, not adaptation in general. Second, the fixed optimal dual's survival under this objective shows per-campaign pricing itself is not the problem, as linear-programming duality predicts: a static optimal price vector can carry the allocation. What fails as contention grows is learning that vector online. This second conclusion holds under the click objective: the revenue results below show a one-shot optimal dual mispricing campaigns when the realized arrival order departs from the fluid plan its prices were solved against. The sharper claim, that prices should be *obtained by solving* and re-solved on the pacing cadence, is stated as the deployment rule below.

The deployment rule follows: the LP's prices are what matter, so obtain them by solving, and recompute them on the pacing cadence (re-solve). Holding a one-shot LP optimum (the fixed dual) suffices under the click objective but is not universally robust, as the revenue results below show; online price learning is competitive only while contention is mild, and no adaptive variant we test survives heavy over-subscription. Re-solving is the single price source positive in every regime we measure.

The regime map reproduces on a second, independent dataset. We repeat the whole protocol on the Outbrain content-recommendation log (87M joined impressions, a slate-click structure unlike Taobao's display, with the segmentation axis fixed in advance to the most persistent one measured on the planning window, page platform, whose interaction residuals correlate at $r = 0.95$ across the shared campaign-by-platform cells; the axis has three platform levels, so this is a statement about

the stability of campaigns' platform preferences, not a many-point fit). At native quotas the Outbrain market is barely contended, per-segment supply dwarfs the quotas, so every policy sits within a percentage point of greedy, exactly the low-contention corner of the map. Under the same over-subscription stress, the ordering returns and sharpens: the re-solved plan is the only policy positive at every level (+0.3 to +0.6%), the static plan and the fixed optimal-dual degrade gently, and online scalar pacing collapses far more steeply than on Taobao, to −19%, −34%, and −44% at 1.5×, 2×, and 3× quotas. That online price learning, not per-campaign pricing, is the failure under contention, and that solving for the prices is the robust choice, holds across two datasets from different domains. Two qualifications attach to this second market. The over-subscription sweep is a counterfactual market that Outbrain's operator never ran: native quotas are barely contended, so the stress levels probe a demand regime absent from the log, and the result is a statement about the mechanism under contention, not about Outbrain's realized traffic. And the native low contention is measurable before deployment: the per-segment supply exceeds the median quota by more than two orders of magnitude, so the abstain-when-uncontended condition is a pre-deployment check (compare quota to segment supply) rather than a judgment made after seeing the lift. The truncated final serving day, whose partial log would understate late-day supply, is excluded throughout.

The regime map also survives a change of objective, from clicks to revenue under guaranteed delivery. We reformulate the market so campaigns carry heterogeneous per-impression values and unmet quotas cost a makegood penalty: half the campaigns, drawn at random, pay per click, so their value on a segment is $\mathrm{cpc}_c \cdot p_{cs}$ and rides on the interaction the paper is about, and half pay per impression at a flat cpm_c that rewards reach and is indifferent to the segment, with the two types calibrated to the same mean value per impression so neither dominates by construction; delivery is capped at the booked quota and each undelivered guaranteed impression costs a penalty β_c. Across eight independent value draws and four contention levels, the static plan and the re-solved plan are the top tier in every one of the resulting cells, positive throughout (about +0.4% to +5% over revenue-greedy at makegood-at-par pricing), and their advantage *grows* as the guarantee hardens: sweeping the penalty from a pure spot market to a near-hard guarantee, the plan-tier lift rises into double digits under heavy contention, because a makegood-exposed seller is punished most for exactly the misallocation that planning removes. The adaptive policies stay positive but trail the plan tier throughout; unlike the pure-click objective they do not collapse to large negatives, because a mispaced impression still earns its flat CPM value and still retires a makegood liability, so the revenue objective is more forgiving of online mispricing than clicks are. The fixed optimal dual is the least stable policy: positive at exact feasibility in every draw (+1% to

+4%) but negative under contention in the majority of them, as low as −7.7%, because a price vector solved once against the fluid plan starves guaranteed campaigns when the realized arrival order departs from it, earning high gross value while paying large makegood penalties, a per-campaign decomposition the released harness prints. Re-solving on the pacing cadence removes the fragility, positive in every draw at every level. The deployment rule, solve for the prices and re-solve on the pacing cadence, is therefore a property of the allocation, not of the click objective it was first measured on.

A battery of robustness checks anchors the table (a user-clustered bootstrap at z = +4.55, an across-fold t-test, a first-exposure fatigue control, a support audit, and a propensity-weighted outcome bracket); none overturns the ordering, and an outcome-grounded estimate on the small logging-matched subset is inconclusive, the inherent limit of public-log evaluation. Appendix A.4 reports them in full.

6 A reference serving architecture

This section describes the deployment design that motivated the study: an operator-grade two-stage architecture that compiles the allocation plan into constant-time lookup serving. It is a design sketch: the experiments of Sections 3–5 validate the allocation core (plan versus serving policy at segment granularity), not the per-subscriber machinery described here.

The design targets the operator's data position. A web ad network observes a browser: a resettable, unverified cookie, activity on the sites it instruments, and demographics inferred by guesswork. A network operator observes a customer: a verified identity anchored in a billing account, declared attributes, financial signals, and, because all browsing crosses the gateway, complete response attribution joined under one stable identifier.

Dimension	Web ad network	Network operator
Identity	Cookie or device fingerprint; resettable, unverified	Subscriber account; stable, verified, one per person
Demographics	Inferred from behavior	Declared at subscription: age, gender, address
Financial signal	None or modeled	Plan, spend level, content purchase history
Location	Coarse, via IP	Network-observed, continuous

Dimension	Web ad network	Network operator
Browsing coverage	Sites the network instruments	All traffic through the gateway
Exposure control	Per-site frequency caps	Global caps across all inventory, per subscriber

The vantage point carries obligations that shape the design. Subscriber data is usable for advertising only within a consent and data-protection framework: profiling requires a lawful basis and per-subscriber opt-out under regimes such as the GDPR and ePrivacy rules, and gateway-observed browsing is subject to purpose limitation. The architecture is built for data minimization under those constraints: raw attributes and browsing records are confined to the offline stage, the serving path receives only anonymous cluster distances and campaign eligibility lists, and every online decision reads aggregate tables rather than individual histories. Segment coarseness is itself a privacy parameter: the segments that carry allocation value in Section 4 are a handful of aggregate behavior tiers, not individual dossiers. Not every subscriber has a profile at all: prepaid subscribers may be anonymous, which is why the clustering machinery supports membership computed from response behavior alone (Appendix A.5).

The full serving design is a two-stage architecture. An offline analytics stage learns the plan's inputs, per-segment traffic forecasts and per-(campaign, segment) click probabilities, from event history, clusters subscribers on behavior, and compiles the resulting plan into compact lookup tables. An online engine then executes one impression at a time in constant time: a seven-step decision reads the pre-computed tables to pick the eligible, quota-remaining campaign whose plan-adjusted score is highest, blends per-cluster click estimates by soft membership, and paces delivery by tracking each campaign against its planned trajectory. Because the offline stage holds all raw attributes while the serving path sees only anonymous cluster distances and eligibility lists, the design realizes the data-minimization posture the operator setting requires. Appendix A.5 gives the full design, its data structures, and the pacing rule; the experiments in this paper exercise the allocation core it compiles, not the per-subscriber serving machinery.

7 Related work

The allocation machinery this paper uses, a transportation LP with dual-price pacing, is standard; its contribution sits in five places that the prior literature leaves open, summarized in the table before the discussion.

Line of work	Prescribes the segmentation	Pre-committed, blind test	Public-log, hardened OPE	Regime map (plan vs. adapt)	Revenue with guarantees
Online matching, AdWords [6, 7]					
Random-permutation online LP [8–10]			~		~
Guaranteed display: forecasts, SHALE [11, 12]					✓
Dual mirror descent [21]				~	✓
Yield optimization, ad exchange [22]					✓
Budget-pacing controllers [13, 14]			~		✓
This paper	✓	✓	✓	✓	✓

✓ addresses the question, ~ partially, blank not addressed. The prior lines take the audience segmentation as given, evaluate on theory or private production data, and do not characterize when a compiled plan beats online adaptation; the guaranteed-delivery and pacing lines share this paper's revenue-with-penalties objective but not its segmentation rule or its public evaluation. The paragraphs below detail each line.

Online matching and allocation. The theoretical core of ad allocation is online bipartite matching, from the optimal $1 - 1/e$ algorithm of Karp, Vazirani, and Vazirani [6] through its budgeted AdWords generalization [7]. Under stochastic arrivals, the random-permutation model admits near-optimal algorithms that learn dual prices from an initial sample [8], and primal-dual training on forecast samples extends to display allocation [9] and general online linear programming [10]. This line of work supplies the adaptive dual-price policy we use as the strong online baseline; our contribution is orthogonal: an empirical account, on public logs, of when a compiled offline plan beats such adaptive methods and what segmentation makes the plan's advantage real.

Guaranteed display allocation. Closest in spirit are systems that precompute compact allocation plans for guaranteed-delivery display advertising: optimal online assignment with forecasts [11] and the SHALE algorithm [12] both compute offline parameters that a lightweight online server

executes, as our architecture does. Those works optimize the plan's representation and solve time; we address the two questions they leave open in deployment: which audience segmentation gives the plan exploitable, durable structure (the persistence rule of Section 4), and how to measure a plan's realized value against strong baselines without an online experiment. Budget-pacing controllers [13, 14] are the production embodiment of the adaptive alternative and correspond to the pacing loop our serving stage carries in Appendix A.5. Relative to this line, our contributions are a pre-committable, falsifiable segmentation test and a public-log evaluation protocol; the architecture itself is a deployment sketch.

Click prediction and offline evaluation. Industrial click-through-rate modeling ranges from large-scale logistic regression [15] and field-aware factorization machines [16] to attention architectures over user behavior [17, 3]. Our click model is deliberately simpler, hierarchical empirical rates with soft cluster membership, because the architecture consumes any calibrated estimator and the allocation question is about the pairing structure, not the estimator frontier. For evaluation, replay methods [18], counterfactual reasoning in computational advertising [19], and offline A/B testing [20] establish the off-policy toolbox; the dual-model scoring of Section 3 applies the same discipline to allocation policies specifically, where we show single-model replay inflates lift through the plan’s own estimation noise.

8 Discussion

Four design principles carry the empirical findings, persistence-chosen segments, solved prices, and a re-solve cadence, into a deployable system.

The optimization lives where time is cheap. Global allocation over all campaigns and segments is a large linear program, and it is solved where minutes are available, not milliseconds. The serving path touches five small tables and does arithmetic. This is the classic plan/execute split, applied so that the online system inherits the quality of the offline optimum at lookup cost.

Pooling defeats sparsity. No individual generates enough clicks to estimate their own response rates. Clusters do. Soft membership then re-personalizes the pooled estimates, blending each subscriber’s several behavioral neighborhoods instead of forcing a single label. The same back-off logic handles new campaigns (borrow from similar campaigns) and profile-less subscribers (cluster by response history alone), so one mechanism covers three cold-start problems.

The plan is a baseline, not a cage. Real traffic never matches the forecast. By carrying planned counts into the serving state and steering with a performance index, the system tolerates forecast error gracefully: small deviations are corrected impression by impression, and persistent ones are evidence, fed back into the next modeling cycle rather than silently accumulated.

Every representation is serving-shaped. Bit masks make targeting tests branchless; pre-computed distances make personalization a weighted sum; per-(cluster, campaign) rows make pacing a two-row query. The offline stage does not merely produce a model; it compiles one, in the same sense a compiler turns a program into a form the machine executes directly.

9 Scope and assumptions

The method is built for operator-owned advertising markets, retail-media networks being the largest contemporary instance and the mobile-operator system that motivated it the originating one, where direct-sold campaigns carry guaranteed impression quotas with makegood penalties, logged-in first-party identity yields audience segments whose campaign-by-segment interaction persists across planning windows, and booked demand meets or exceeds supply on those segments; where any of these conditions fails, in particular where supply dwarfs demand, planning offers little over serving greedily. The conditions are checkable before deployment, and the paper measures each: interaction persistence $r \geq \sim 0.4$ on the chosen axis (Section 4), per-cell planning density near 10^3 (Algorithm 1), a re-plan cadence at least as fast as portfolio churn (the Avazu boundary), and contention κ_s at or above one on the interaction-carrying segments (Algorithm 1, step 6). The three public datasets are positioned accordingly: Taobao is the in-vertical evidence where all conditions hold, Avazu is the churn boundary where a daily plan fails until the cadence shrinks, and Outbrain is the contention boundary where an uncontended market yields little until demand is stressed. The two boundaries are the edges that make the deployment claim falsifiable.

The architecture optimizes expected revenue for a known campaign portfolio over a planning horizon; it does not address auction-based pricing, real-time bidding against external demand, or creative selection within a campaign. Its guarantees rest on the stability assumptions of Appendix A.5: allocation quality degrades in proportion to how far future traffic and response distributions drift from the modeled ones between refinement cycles, which sets the practical requirement for how frequently the offline stage must re-run. Exposure policy enforcement is per-subscriber and rule-based; richer fatigue models would slot into step 2 without changing the

surrounding machinery. Finally, window-fitted planning presumes a campaign portfolio stable over the planning horizon, which is the operator's contracted-campaign setting; in marketplaces where the ad portfolio and its delivery mix turn over within a day, the re-planning interval must shrink to the churn timescale for the plan to retain value.

References

1. Q. Huangfu and J. A. J. Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018.
2. Alibaba Tianchi. Ad Display/Click Data on Taobao.com. Dataset 56, tianchi.aliyun.com/dataset/56, accessed August 2026.
3. Y. Feng, F. Lv, W. Shen, M. Wang, F. Sun, Y. Zhu, and K. Yang. Deep Session Interest Network for Click-Through Rate Prediction. In *Proceedings of IJCAI*, pages 2301–2307, 2019.
4. Avazu. Click-Through Rate Prediction. Kaggle competition, kaggle.com/c/avazu-ctr-prediction, 2014, accessed August 2026.
5. A. Mehta. Online Matching and Ad Allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.
6. R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd ACM Symposium on Theory of Computing (STOC)*, pages 352–358, 1990.
7. A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. AdWords and generalized online matching. *Journal of the ACM*, 54(5), Article 22, 2007.
8. N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, pages 71–78, 2009.
9. J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. In *Proceedings of the 18th European Symposium on Algorithms (ESA)*, pages 182–194, 2010.
10. S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.
11. E. Vee, S. Vassilvitskii, and J. Shanmugasundaram. Optimal online assignment with forecasts. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*, pages 109–118, 2010.
12. V. Bharadwaj, P. Chen, W. Ma, C. Nagarajan, J. A. Tomlin, S. Vassilvitskii, E. Vee, and J. Yang. SHALE: an efficient algorithm for allocation of guaranteed display advertising. In *Proceedings of the 18th ACM SIGKDD Conference (KDD)*, pages 1195–1203, 2012.
13. D. Agarwal, S. Ghosh, K. Wei, and S. You. Budget pacing for targeted online advertisements at LinkedIn. In *Proceedings of the 20th ACM SIGKDD Conference (KDD)*, pages 1613–1619, 2014.

14. J. Xu, K.-c. Lee, W. Li, H. Qi, and Q. Lu. Smart pacing for effective online ad campaign optimization. In *Proceedings of the 21st ACM SIGKDD Conference (KDD)*, pages 2217–2226, 2015.
15. H. B. McMahan, G. Holt, D. Sculley, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD Conference (KDD)*, pages 1222–1230, 2013.
16. Y.-C. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin. Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys)*, pages 43–50, 2016.
17. G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai. Deep Interest Network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD Conference (KDD)*, pages 1059–1068, 2018.
18. L. Li, W. Chu, J. Langford, and X. Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the 4th ACM WSDM Conference*, pages 297–306, 2011.
19. L. Bottou, J. Peters, J. Quiñonero Candela, D. X. Charles, M. Chickering, E. Portugaly, D. Ray, P. Y. Simard, and E. Snelson. Counterfactual reasoning and learning systems: the example of computational advertising. *Journal of Machine Learning Research*, 14(1):3207–3260, 2013.
20. A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé. Offline A/B testing for recommender systems. In *Proceedings of the 11th ACM WSDM Conference*, pages 198–206, 2018.
21. S. Balseiro, H. Lu, and V. Mirrokni. Dual mirror descent for online allocation problems. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 613–628, 2020.
22. S. Balseiro, J. Feldman, V. Mirrokni, and S. Muthukrishnan. Yield optimization of display advertising with ad exchange. *Management Science*, 60(12):2886–2907, 2014.
23. M. Dudík, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1097–1104, 2011.
24. Y.-X. Wang, A. Agarwal, and M. Dudík. Optimal and adaptive off-policy evaluation in contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3589–3597, 2017.

Appendix

A.1 Worked example: the cost of greedy serving

Consider the smallest instance in which greedy serving and planned allocation come apart. Two campaigns run at the same time: a soft drink and a sports car. Each has bought exactly **two impressions**. Four subscribers will each generate one ad opportunity. From past behavior we can estimate every subscriber's probability of clicking each campaign:

Subscriber	P(click | drink)	P(click | car)
u_1	0.10	0.80

Subscriber	P(click | drink)	P(click | car)
u_2	0.30	0.50
u_3	0.20	0.60
u_4	0.40	0.70

A greedy server shows every subscriber their personally best ad. All four prefer the car, so the car campaign's quota is consumed by whoever happens to arrive first, say u_1 and u_2. The drink campaign is then forced onto the remaining two:

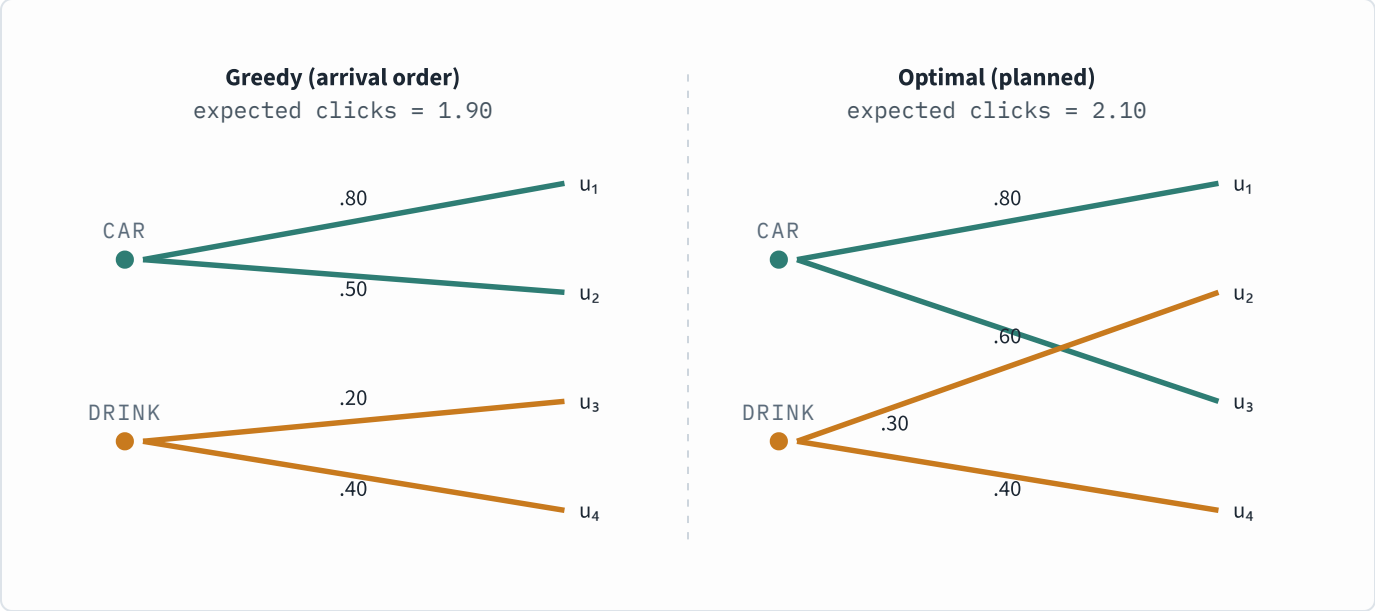

Figure 2. Two campaigns, four subscribers, two impressions per campaign. Greedy serving fills the car campaign with the first arrivals (u_1, u_2) for 0.80 + 0.50 + 0.20 + 0.40 = 1.90 expected clicks. The optimal assignment gives the car campaign u_1 and u_3 and the drink campaign u_2 and u_4, for 0.80 + 0.60 + 0.30 + 0.40 = 2.10, an 11% lift from re-pairing the same impressions. The greedy value depends on arrival order; averaged over all orders it is 1.80, so the drawn order is a conservative case.

Nothing about the subscribers or the campaigns changed between the two panels. The only difference is that the right-hand assignment was chosen *jointly*: the car campaign gives up u_2 (a 0.50 click chance), whose 0.30 with the drink remains a strong pairing, and takes u_3 (0.60) instead. Each individual swap looks locally worse for someone; the total is strictly better.

The gap grows with scale. With hundreds of campaigns, millions of subscribers, targeting constraints, frequency caps, and time-of-day windows, greedy serving does not just lose a few percent: it systematically starves narrow-audience campaigns, over-exposes indifferent

subscribers, and leaves contracted quotas unfilled at the end of the flight. Optimal allocation is not a refinement of ad serving; it is the difference between selling inventory and wasting it.

> **The core insight:** click probability is a property of a *pair* (audience, campaign), not of a campaign alone. Revenue is maximized by choosing the pairing globally, subject to each campaign's quota, and this choice can be made ahead of time because audience behavior is statistically predictable even though individual events are not.

A.2 Proof of interaction invariance

Proof. Expanding the objective, $V(x) = \Sigma_c\, a_c\, (\Sigma_s\, x_{c,s}) + \Sigma_s\, b_s\, (\Sigma_c\, x_{c,s}) + \Sigma_{c,s}\, x_{c,s}\, g_{c,s}$. The first sum depends on x only through the campaign margins, the second only through the segment margins; both are held fixed, so they are equal for x and x' and cancel on subtraction. Only the interaction term remains. □

A.3 Simulation as calibration

The regime map of Section 5 was measured by a replay harness; this section calibrates that harness on a fully synthetic market where ground truth is known and every quantity is controlled. The four-subscriber example of Appendix A.1 proves the mechanism exists; a Monte-Carlo simulation quantifies it at realistic scale. Each replication generates a synthetic market: 60 segments with lognormal traffic (about 200,000 impression opportunities), 25 campaigns whose quotas sell 80% of expected inventory, per-impression prices drawn uniformly on [$0.05, $0.30] and per-click prices on [$0.50, $3.00], and click probabilities from a campaign-audience affinity model (each campaign genuinely fits some audiences better than others, with rates spanning 0.1%–8%). Each campaign is eligible, by targeting, on roughly 60% of segments.

Two policies then serve the *identical* event stream under identical quotas, prices, probabilities, and targeting. The **greedy** policy is the conventional per-event rule: serve the eligible, quota-remaining campaign with the highest immediate expected revenue. The **planned** policy solves the linear program of Section 2 offline on the traffic forecast, then serves each event by drawing down the planned allocation, falling back to greedy where the plan is locally exhausted. The plan never sees the realized traffic; events are drawn randomly around the forecast, so the planned policy faces genuine forecast error. A third quantity, the **hindsight bound**, is the offline optimum computed on the traffic that actually occurred; no online policy can exceed it.

Measure (20 replications)	Mean	Range
Revenue lift, planned over greedy	+6.5%	+4.1% to +8.8%
Expected-click lift, planned over greedy	+33.0%	+22.9% to +40.6%
Planned revenue as share of hindsight bound	93.8%	≥ 91.7%
Greedy revenue as share of hindsight bound	88.1%	≥ 85.4%
Quota fill, both policies	100%	all replications

Figure 3. Revenue captured by each policy as a share of the hindsight offline optimum, per replication. Planned allocation dominates greedy serving in all 20 replications; the two distributions do not overlap (planned minimum 91.7%, greedy maximum 91.6%).

Both policies fill every quota, so the entire gap is *pairing quality*: the same impressions, sold to the same campaigns, assigned to different audiences. This also explains why the click lift (33%) exceeds the revenue lift (6.5%): per-impression fees are earned identically by both policies once quotas fill, so planning's advantage is concentrated wholly in the click-priced component of revenue. In markets weighted further toward performance pricing, the revenue gap widens toward the click gap.

The simulation carries its own controls. Both policies are verified against the hindsight bound in every replication. On control instances where all campaigns share identical click probabilities, so pairing cannot matter, the measured lift is 0.000%, confirming that the gap measures allocation quality and not an artifact of the harness. And the planned policy's 93.8% of a bound computed

with perfect hindsight shows that forecast noise costs the plan only a few points, the residual that Appendix A.5's pacing mechanism and the refinement loop exist to absorb.

A.4 Extended robustness checks

Three robustness checks anchor the table. The headline planned-versus-greedy comparison survives a user-clustered bootstrap (z = +4.55) and an across-fold t-test (t(4) = +5.50); at exact feasibility the strong policies are statistically close, so the ordering across contention regimes, not any single feasibility cell, is the claim. Re-pairing the impression stream changes per-user exposure composition; re-scoring under a first-exposure table (each user's first impression of each campaign only, which removes repeat-exposure composition) moves the headline from +3.46% to +3.88%, so exposure fatigue does not manufacture the result. The regime map is a single-market result: the over-subscription sweep is run on the Taobao top-100 reconstruction, while Avazu contributes only the stability boundary. The z-values in the table above the headline row come from the parametric binomial model; the conclusions that rest on them (the re-solved plan's dominance at every over-subscribed scale, the dual collapse) hold with the same sign and comparable magnitude under the user-clustered bootstrap where we recomputed them. The remaining, and inherent, limitation is that these are model-scored expected clicks: an outcome-grounded estimate is possible only on the 2–2.4% of impressions where a policy's choice coincides with the logged action, a subset both tiny and selected toward the logging policy. On it, a self-normalized inverse-propensity estimate with an estimated segment-level logging propensity puts planned within ±1pp of greedy at t(4) = −0.5, that is, inconclusive; the matched-subset calibration shows the model scorer understates realized CTR slightly and symmetrically for both policies (about 6.8% realized versus 6.5% modeled), so it is not inflating the gap. Public logs cannot close this gap; only an interventional deployment can.

A.5 Reference serving architecture

This appendix gives the full deployment design summarized in Section 6: the offline stage that compiles the plan and the online stage that executes it per impression. It is a design sketch; the experiments of Sections 3–5 validate the allocation core it compiles, not the per-subscriber serving machinery detailed here.

offline stage: learning the inputs to the plan. The allocation program needs two inputs that do not exist in raw data: predicted traffic per segment (H_s) and click probabilities per campaign-

segment pair ($p_{C,S}$). The offline analytics stage builds both from event history, under three modeling assumptions, each holding with high probability rather than certainty:

1. The distribution of (context, profile) pairs observed in the past will recur in the future.
2. Subscribers who responded similarly under similar conditions will continue to respond similarly.
3. Subscribers who respond similarly share something observable in their profiles or histories.

Assumption 1 justifies forecasting traffic from history; assumptions 2 and 3 justify pooling sparse per-subscriber click data into clusters that have enough volume to estimate probabilities reliably.

records. The atomic unit of evidence is an ad response record: who saw what, in what context, and whether they clicked.

Time of day	Day of week	Content category	Campaign	Click
`11:23`	Monday	News	`C1 (active)`	`0`
`14:15`	Monday	Sport	`C2 (active)`	`1`
`18:14`	Sunday	Sport	`C4 (past)`	`1`

Records are grouped per subscriber and joined with the profile (age, gender, location, income band, purchase indicators) where one is available. Past campaigns matter as much as active ones: a subscriber's reaction to a finished campaign is evidence about how they will react to similar future ones.

and clustering. Subscribers are clustered by a weighted distance that mixes profile similarity with response-pattern similarity: response to the same campaigns (active and past) and to similar campaigns. Each attribute contributes its own distance component:

Attribute	Distance component
Age	Absolute difference
Gender	Fixed penalty if different
Campaign affinity	By campaign category; fixed penalty if different
Day of week	Absolute difference, circular
Location (x, y)	Straight-line distance

Membership is deliberately **soft**. A subscriber is not filed into one cluster; they carry a pre-computed distance to each nearby cluster. A 35-year-old married streaming subscriber in a large city may sit at distance 0.8 from the “metropolitan professionals” cluster and 0.1 from “music lovers”; a 22-year-old gaming subscriber in the same city sits close to “gamers” and “music lovers” instead. These distances become interpolation weights at serving time (Appendix A.5), so predictions blend the behavior of every cluster the subscriber resembles.

representation and click estimation. Each cluster is summarized by two compact structures. Its *context slots* describe when and where the cluster generates traffic, with predicted hit counts; this is the supply side H_s. Its *click statistics* record impressions and clicks per campaign per context slot, the raw material for $p_{c,s}$:

Context slot	Time of day	Day of week	Category	Predicted hits
CS1	10:00–12:00	Mon–Tue	News, Sport	3,000
CS2	11:00–17:00	Sun	Sport	4,000

Campaign	Context slot	Impressions	Clicks
C1	CS1	1,000	3
C1	CS2	2,000	2
C2	CS1	1,000	7

For a brand-new campaign with no history of its own, probabilities are borrowed from the most similar past or active campaigns in the same category, the same pooling idea applied across campaigns instead of across subscribers. Where the profile itself is missing (anonymous or prepaid subscribers), distance to clusters is computed from response history alone, and the subscriber inherits the response statistics of the cluster they behave like. The design degrades gracefully as data thins out: per-subscriber evidence backs off to cluster evidence, and per-campaign evidence backs off to category evidence.

the plan into lookup tables. Solving the allocation program yields planned impression counts per (campaign, cluster, context slot). Everything the online server needs is then flattened into three lookup structures, so that no clustering, no distance computation, and no optimization happens on the serving path:

Structure	Keyed by	Contents
Model table	`cluster × campaign`	Targeted time slots and content categories as bit masks; planned impressions; expected clicks
Match table	`subscriber`	Campaigns this subscriber is eligible for under targeting rules
Cluster table	`subscriber`	Distance to each nearby cluster (the soft-membership weights)

Time and content targeting are encoded as fixed-width bit masks, 168 weekly hour-slots and up to 256 content categories, so an eligibility test at serving time is a bitwise AND rather than a rule evaluation. A model table row reads, in full: *"cluster L1, campaign C1, time mask 0100…, content mask 1111…, 10,000 planned impressions, 450 expected clicks."* That one row carries targeting, allocation, and the performance baseline the online stage will pace against.

online stage: executing the plan in real time. At serving time an impression event arrives carrying a subscriber identifier and a context (time, day, content category of the page being viewed). The decision engine must return a single campaign within the latency budget of a page load. It consults the three model structures plus two live state tables: a **status table** mirroring the model table but accumulating *actual* impressions and clicks, and a **response table** holding each subscriber's recent exposure records.

Figure 4. The closed loop. Offline analytics compiles the model and allocation plan into lookup tables; the online engine executes them per impression and streams response records back, which the next analytics cycle uses to refine clusters, probabilities, and the plan.

The seven-step decision.

1 **Find eligible campaigns**

`match table` Look up the campaigns this subscriber may be shown under targeting rules.

2 **Apply exposure policy**

`response table` Drop campaigns that would violate frequency rules for this subscriber, a minimum interval between exposures, a maximum number of exposures per day.

3 **Fetch cluster distances**

`cluster table` Retrieve the subscriber's pre-computed distances to nearby clusters.

4 **Predict click probability per candidate**

`model table` Blend cluster-level click rates with the soft-membership weights (Appendix A.5).

5 **Compute pacing per candidate**

`status table` Compare actual delivery against plan, prorated to this point in the flight (Appendix A.5).

6 **Decide**

If any campaign is critically underserved, serve it. Otherwise drop overserved campaigns and serve the candidate with the greatest expected revenue.

7 **Update state**

`status + response tables` Increment the served campaign's actual counters and append the exposure record, which later feeds back to analytics.

Every step is a keyed lookup, a bit-mask test, or arithmetic over a handful of rows. The expensive work (clustering, probability estimation, optimization) was all paid for offline.

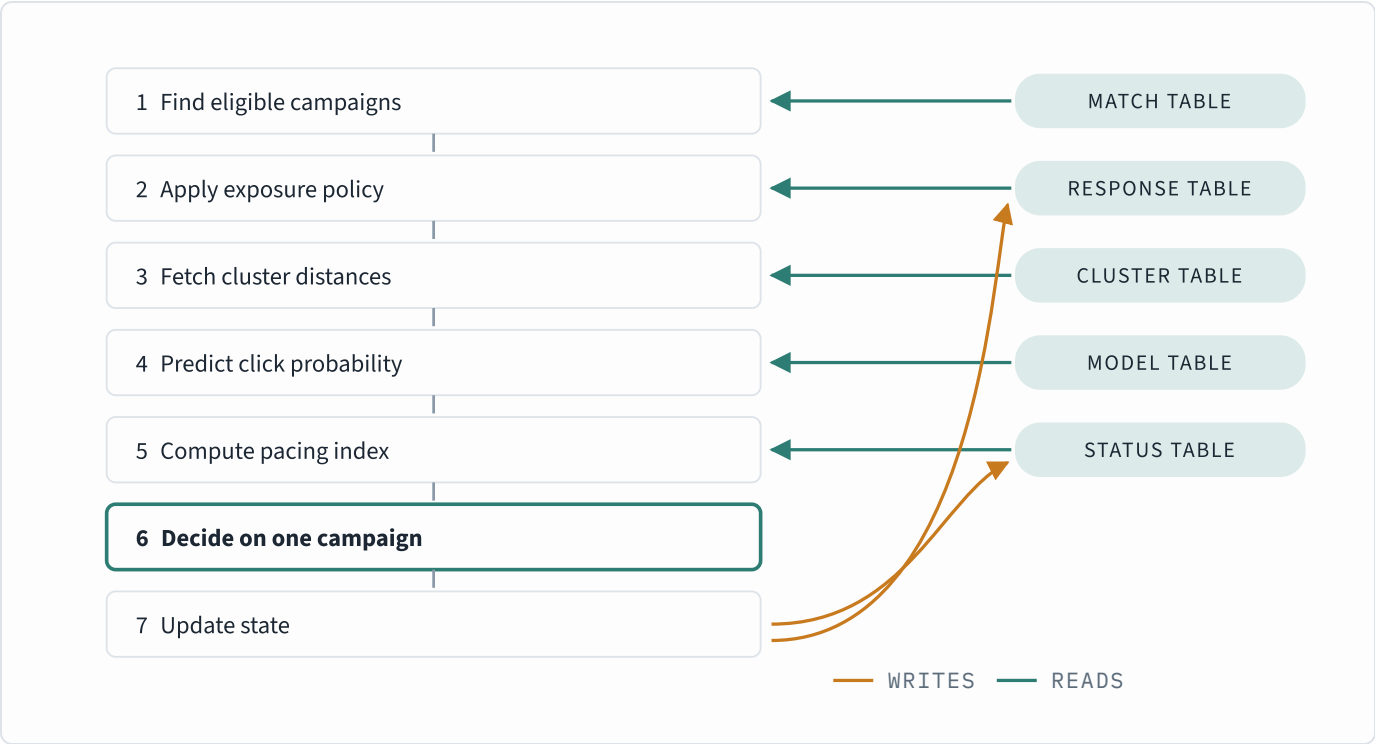

Figure 5. The seven-step decision path and its data dependencies. Steps 1–5 each read one pre-computed or live table; step 6 is pure arithmetic over the candidates; step 7 writes the served impression back into the status and response tables, which is also the record that feeds the next offline refinement cycle.

prediction by soft membership. Suppose the event context is *Monday, 14:00, Sport* and the subscriber sits at weight 0.2 to cluster L1 and 0.8 to cluster L2 (weights derived from the stored distances). The model table gives campaign C1's historical rates in the matching context: 40 clicks per 8,000 impressions in L1 and 60 per 10,000 in L2. The prediction is the weighted blend:

```
p(click | C1)  =  0.2 · (40 / 8,000)  +  0.8 · (60 / 10,000)  =  0.0058
```

Because the subscriber borrows from every cluster they resemble, the estimate is stable even for subscribers with thin personal histories, and it is computed from two table rows and four multiplications.

Pacing: planned versus actual. The plan is a trajectory, not just a total. If a campaign is one third of the way through its flight in a given cluster, it should have received roughly one third of its planned impressions there. The engine aggregates this across the clusters where the campaign runs:

Cluster	Flight elapsed	Planned by now	Actual
L1	33%	8,000 × 0.33 = 2,640	2,000

Cluster	Flight elapsed	Planned by now	Actual
L2	50%	10,000 × 0.50 = 5,000	5,100
Total		**7,640**	**7,100**

```
performance index  =  actual / planned-by-now  =  7,100 / 7,640  ≈  93%
```

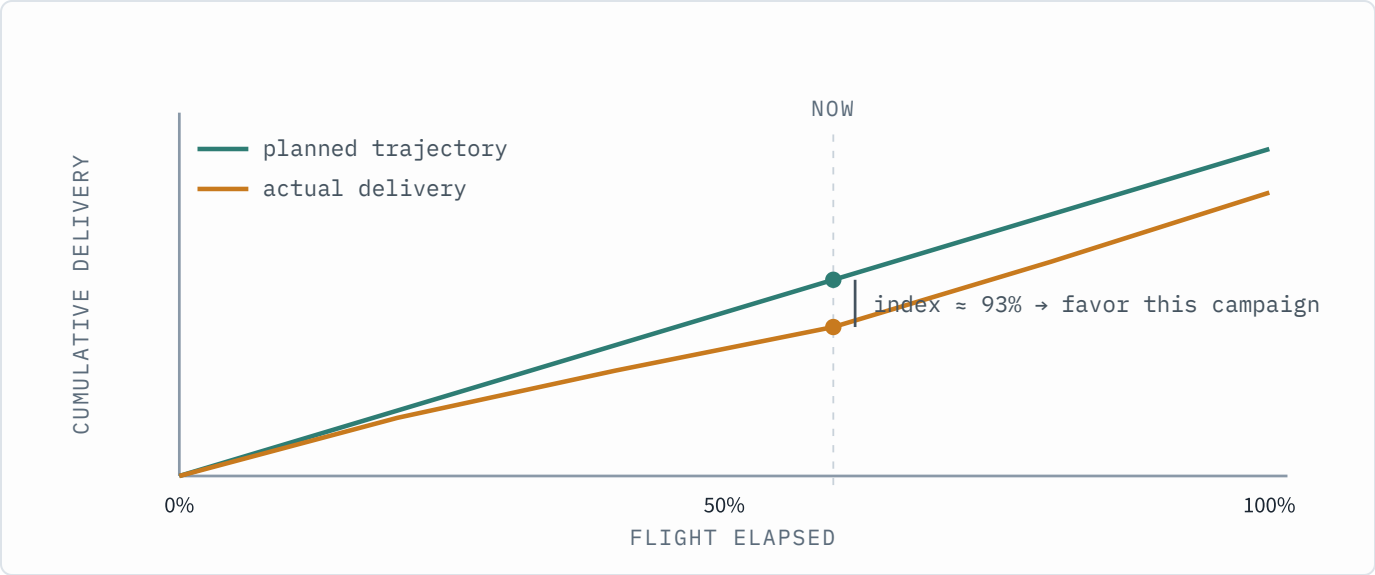

Figure 6. Pacing as trajectory tracking. The plan implies a cumulative delivery path; actual delivery drifts below it, the performance index measures the gap at the current moment, and the decision rule favors the lagging campaign on suitable impressions until the curves reconverge.

An index near 100% means the campaign is on plan. Far below a threshold (say 50%), the campaign is critically underserved and jumps the queue (step 6); far above another (say 120%), it is overserved and stands aside. Between the thresholds, campaigns compete on the same per-impression value the plan optimizes, expected revenue = impression price + click price × P(click):

Campaign	Impression price	Click price	Perf. index	P(click)	Expected revenue
C1	$0.10	$1.00	55%	0.01	$0.110
C2	$0.10	$0.80	91%	0.03	$0.124
C3	$0.10	$1.50	96%	0.02	$0.130
C4	$0.10	$1.00	128%	0.07	$0.170

C4 carries the greatest expected revenue but sits at 128% of plan, past the overserved threshold, so it is dropped; C1 at 55% is behind plan but above the critical threshold, so it earns no override. Among the remaining candidates the engine serves C3, the greatest expected revenue for this impression. The pacing mechanism is what lets a static offline plan survive contact with real, noisy traffic: the plan sets the destination, and the index steers thousands of small per-impression corrections toward it. Deviations that pacing cannot absorb (a traffic forecast that proves wrong, a campaign that outperforms its category prior) are exactly what the feedback loop carries back into the next planning cycle.

Preprint, August 2026. Code and replay harness: github.com/ApartsinProjects/optimal-impression-allocation.