Ad Serving with Multiple KPIs

Brendan Kitts
PrecisionDemand
Seattle, USA
bkitts@gmail.com

Michael Krishnan,
Ishadutta Yadav, Yongbo Zeng,
Garrett Badeau
Adap.tv
San Mateo, USA

Andrew Potter,
Sergey Tolkachov,
Ethan Thornburg,
Satyanarayana Reddy Janga
AOL Platforms
San Mateo, USA

ABSTRACT

Ad-servers have to satisfy many different targeting criteria, and the combination can often result in no feasible solution. We hypothesize that advertisers may be defining these metrics to create a kind of "proxy target". We therefore reformulate the standard ad-serving problem to one where we attempt to get as close as possible to the advertiser's multi-dimensional target inclusive of delivery. We use a simple simulation to illustrate the behavior of this algorithm compared to Constraint and Pacing strategies. The system is then deployed in one of the largest video ad-servers in the United States and we show experimental results from live test ads, as well as 6 months of production performance across hundreds of ads. We find that the live ad-server tests match the simulation, and we report significant gains in multi-KPI performance from using the error minimization strategy[1].

CCS CONCEPTS

• **Information systems → Computational advertising**; • *Applied computing → Online auctions*

KEYWORDS

Advertising, targeting, viewability, prediction, optimization, IAB, Internet Advertising Bureau

1 INTRODUCTION

A common problem for ad-servers is to deliver as much value as possible within budget [10], [13]. However, in practice, advertisers routinely add a wide range of other "Key Performance Indicators" (KPI) that the campaign must meet. For example, the Internet Advertising Bureau (IAB) in 2014 has introduced an industry standard, that impressions should be at least 70% viewable for billing to occur [8]. Advertisers may also request that at least 50% of impressions for which they're charged be in the preferred age-gender category. Levels of bot activity usually need to remain below a particular threshold such as 5%. Advertisers may also require that the ad be viewed to completion at least 50% of the time.

These KPIs are usually handled in practice by adding them as constraints to the ad-serving problem. However in many cases, the desired combination of key performance indicators may either be infeasible, or so severely restrict delivery, as to afford little reason to engage with the overhead of running a campaign.

We propose an alternative framework in which the objective is to minimize the vector of constraint error. The resulting system appears to show good behavior on real-world ad-serving problems in which multiple KPIs are common. We test the system in simulation as well as real auction conditions.

2 CANONICAL AD SERVING PROBLEM

Consider an advertiser that has a budget B and wishes to spend it on an ad auction across T discrete periods of time. There are I_t impressions in each period t. Given an offered bid price ${b_{i,t}}^*$ for an incoming impression, the advertiser will "win" the impression at a rate given by $W({b_{i,t}}^*)$ and assuming a Generalized Second Price auction, be charged a clearing price equal to the second bidder's bid plus 1 penny, $b_{i,t} = 0.01 + b_{j,t} : \max b_{j,t} \le {b_{i,t}}^* : j \neq i$. The probability of the impression producing KPI event k will equal ${v_{i,t}}^k$. The KPI rate and budget required per time period are ${V_t}^k$ and B_t, with B_0 indicating the original rate required prior to the start of bidding at time $t = 0$. The task for the ad-server is to set offering bid prices ${b_{i,t}}^*$ for every impression such that the value events ${v_{i,t}}^k$ are maximized. The definition for this problem has been described in much prior work [1, 7, 9]:

$$ {b_{i,t}}^* : \mathbf{max} \sum_{t=1}^{T} \sum_{i=1}^{I_t} W({b_{i,t}}^*) \cdot {v_{i,t}}^k \quad (1) $$

where the spend is lower than a budget constraint.

$$ \sum_{t=1}^{T} \sum_{i=1}^{I_t} W({b_{i,t}}^*) \cdot b_{i,t} \le B \quad (2) $$

Ad-servers also typically include a "smooth delivery" constraint [23, 31]. It is usually considered unacceptable for an ad-server to expend all of its budget in the first hour, and so some constraints are added to ensure that budget is spread over the full set of time periods:

KDD '17, August 13-17, 2017, Halifax, NS, Canada.

ACM ISBN 978-1-4503-4887-4/17/08…$15.00
http://dx.doi.org/10.1145/3097983.3098085

$$\left|\sum_{i=1}^{I_t} W(b_{i,t}^*) \cdot b_{i,t} - B_t\right| \le J (3)$$

Most ad-servers support "performance criteria" that the price paid per event not exceed an advertiser-defined CPA^k price such as Cost Per Click (CPC), Cost Per Acquisition (CPA), Cost Per Viewable (CPV), or Cost per Dollar generated (ROAS) [11, 13]. We define that as follows:

$$\frac{\sum_{t=1}^{T}\sum_{i=1}^{I_t} W(b_{i,t}^*) \cdot b_{i,t}}{\sum_{t=1}^{T}\sum_{i=1}^{I} W(b_{i,t}^*) \cdot v_{i,t}^{k}} \le CPA^k \quad (4)$$

Advertisers routinely add additional KPI requirements for their campaigns including Viewability Rate, Completion Rate, In-Target Rate, Clickthrough Rate, and so on. We therefore have constraint equations [6]:

$$\frac{\sum_{t=1}^{T}\sum_{i=1}^{I_t} W(b_{i,t}^*) \cdot v_{i,t}^{k}}{\sum_{t=1}^{T}\sum_{i=1}^{I_t} W(b_{i,t}^*)} \ge V^k \quad (5)$$

3 STANDARD CONTROL SOLUTION

There have been several published solutions to the above problem [1,10,13,14,17,23,31,32]. In general the problem is approached as a control systems problem (Figure 9) where the plant is considered the auction and bid price is the actuator.

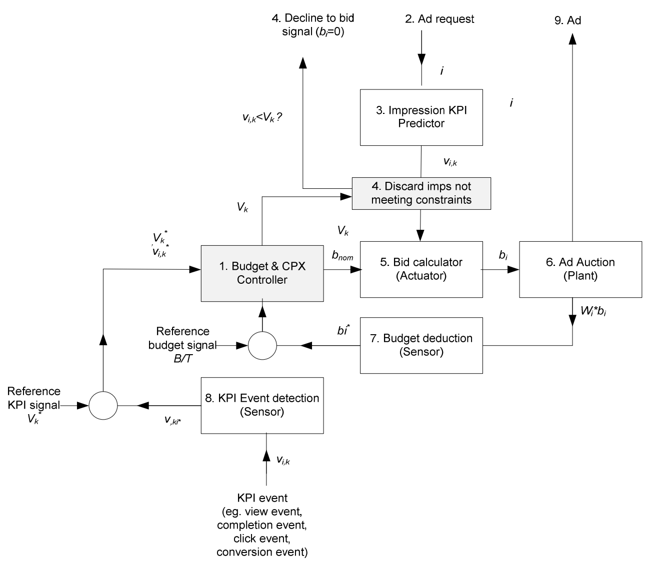

Figure 9: Standard advertising control system.

The system executes the following control loop:

Step 1: Receive a request for ads (Figure 9-2): Let there be a request *i* from a publisher for ads.

Step 2: Evaluate K "impression valuation models" (Figure 9-3): Predict the probability of the impression producing any of the *K* events that the advertiser is interested in $v_i^{k^*}$ (5) such as viewability, click and conversion [13, 21].

Step 3: Filter Traffic not meeting Constraints (Figure 9-4): If the impression fails any of the requirements defined by the advertiser (equation 6) then discard the traffic by setting bid price to zero:

$$\text{If } \exists k: v_i^k < V^k \text{ then } b_i^* = 0 \quad (6)$$

Step 4: Calculate the bid price required for smooth delivery (Figure 9-5): Let b_i^P be the bid such that the expected spend will be as close as possible to the desired spend B_t. Some authors do this by setting a participation rate [1, 10, 17, 31, 32]. Other authors [13] set the bid price directly to throttle. In both cases, the decision variable ultimately is factored into the bid price. The approaches for estimating bid also vary from direct auction modeling [13] to MIMD controllers [33]. In this paper we present a direct modeling approach. Let $W(b_i) = M(b_{i,}, \theta, t)$ be a function mapping the bid price, time, and parameters, to the expected probability of win, and I_t^* a prediction of the number of impressions in this time period. We can select the bid price that minimizes the difference below:

$$b_i^P = b_i : \min|b_i \cdot I_t^* \cdot M(b_i, \theta, t) - B_t| \quad (8)$$

Step 5: Calculate the maximum bid price b_i for achieving the CPA control signal (Figure 9-5): In general bid price is calculated by setting bid price as below; this is called "CostMin" by Karlsson [9]:

$$b_i^k = v_i^{k^*} \cdot CPA_t^k = (v_i^{k^*}/V_t^k) \cdot b_i^P (9)$$

Step 6: Set final bid to the lower of the pacing price and the KPI bid price: This is required due to the nature of the constraint boundaries: if $b_i^k > b_i^P$ then this will drop the expenditure to the pacing price. If $b_i^P > b_i^k$ then $b_i^k$is already at the CPA limit per equation (4), and so increasing the bid further is impossible since it would violate the CPA constraint. This is "a feature – not a bug" of using constraints.

$$b_i^* = \min(b_i^k, b_i^P) \quad (9.2)$$

Step 7: Submit bid price to the auction (Figure 9-6)

Step 8: Deduct the budget (Figure 9-7) and update the KPI counters (Figure 9-8): If the ad's bid was successful in winning the auction, then deduct the clearing bid price b_i from the ad's budget $B = B - b_i$. If an external KPI event is detected then accrue the KPI counters $V^k = V^{k'}+1$.

Step 9: Update the control targets including (Figure 9-1 and 4-2): Update the new control variables, Budget B_{t+1}, Constraint goals $CPA^k{}_{t+1}$ and KPI targets $V^k{}_{t+1}$. A PI Controller can be defined per below for recent time periods as well as all time periods [32]. Karlsson [10] use an alternative approach of deriving full control system plant equations. However this approach requires a fixed analytic function for impressions. Real-time bidding exchange inventory is volatile, and so the model-less PI control approach is more commonly used.

$$V_{t+1}{}^k = \frac{\sum_{\tau\in 1..T} I_\tau \cdot V_\tau^k - \sum_{\tau\in 1..t} I_\tau \cdot V_\tau^k}{I_{t+1}}; \quad (10)$$

$$B_{t+1} = \frac{B - \sum_{\tau\in 1..t} I_\tau \cdot B_\tau}{I_{t+1}}; I_{t+1} = \frac{I^* - \sum_{\tau\in 1..t} I_\tau}{T-t}$$

4 REFORMULATING THE PROBLEM

The problem with using the standard problem definition is that when realistic advertiser KPI constraints are incorporated such as Viewable Rate > 70% and Clickthrough Rate > 1%, it often leads to no feasible solution (Figure 1). It is useful to step back and try to understand why these constraints are being added in the first place. Why would advertisers need to specify a "laundry list" of constraints anyway? If the advertiser is trying to obtain acquisitions, for example, what care they, for the bot rate, viewability rate, completion rate, or any of the other KPIs?

There are several real-world factors that are driving advertisers to specify KPIs: Firstly standards are now being used by the industry that mandate that these are achieved for the traffic to be billable. There is now a 70% viewability requirement [8].

Secondly, this may be an unavoidable product of the advertisers solving a complex estimation problem. Advertisers ultimately want conversion events, but estimating the probability of purchase on each impression may be sparse, custom, or even not practical on the advertising platforms that they're using. They may therefore need to use "high velocity" key performance indicators (KPIs) that are exposed by the ad-server as a "proxy" for what the economically valuable event that they are trying to generate. As a result, the multiple KPIs are used by the advertiser to describe the kind of traffic that they believe have a high probability of purchase.

In this paper we propose a new formulation which is both feasible, and where the objective is to minimize constraint error across multiple KPIs. In order to do this, we propose a modification to the standard control system (Figure 9): the addition of a new component that we call a *KPI Controller* (Figure 10-1). This mechanism will attempt to calculate a bid price that minimizes error *over the vector of KPIs*. The KPI Controller will attempt to keep the performance of the KPIs as close as possible to their reference signal of the multi-dimensional KPI signal that the advertiser has defined as their target.

After adding the KPI Controller to maintain KPIs close to the advertiser's target, we will also remove the hard constraint step that just discarded traffic if it failed to meet the KPI targets (Figure 9-3). The steps of the new control system become:

1. Receive a request to deliver an ad.
2. Execute the K valuation models to predict the probability of this impression eliciting any of the KPI events that are of interest to the advertiser.
3. Don't hard filter the impressions – allow them to be priced (next step).
4. Calculate a bid price that minimizes the multi-KPI error.
5. Submit bid price to auction.
6. Deduct the Budget if the ad wins the auction
7. Update the KPI if an external event is detected.
8. Calculate new KPI and Budget control targets.

The key modification is the bid calculation algorithm. We describe how that works in the next section.

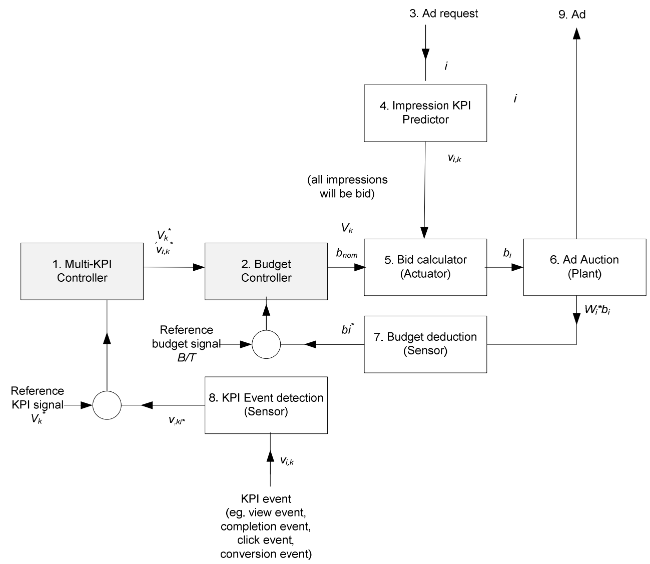

Figure 10: Proposed Multi-KPI control system.

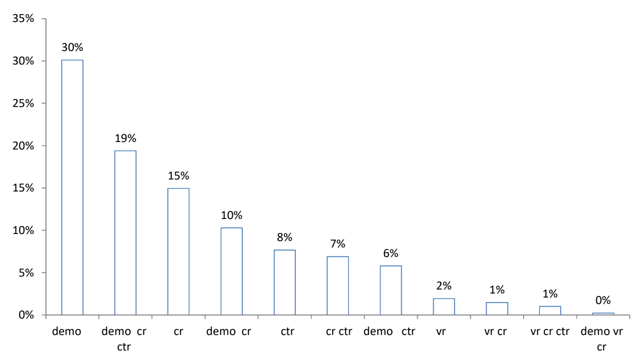

Figure 1: Probability of KPI combination being selected by One Video advertisers

5 ERROR MIN

5.1 Problem Definition

Let us define constraint error ${\Delta_i}^k$ as a measure of the difference between the advertiser's desired KPI ${V_0}^k$ and the current KPI required ${V_t}^k$ during the current time period t.

$$ {\Delta_i}^k = f\left({V_0}^k, {V_t}^k\right) \quad (12) $$

The objective for the optimizer will be to set bid prices such that the constraint error across all KPIs is minimized.

$$ {b_i}^*: \mathbf{min}\, Err = \sum_t^T \sum_i^{I_t} \sum_k^K \mathrm{u}^k \cdot {\Delta_i}^k \quad (13) $$

where $\mathrm{u}^k \in [0..1]: \sum_k \mathrm{u}^k = 1$ are user-defined weights on the KPI errors. The reader should assume these are $\mathrm{u}^k = 1/K$ unless otherwise stated. Let us also define bid prices for Pacing ${b_i}^P$ and CPA ${b_i}^k$ as they are defined in (8) and (9).

5.1 Rate KPI Bid Prices

We now add bid prices for Rate variables – which was the traffic that we were previously discarding (Figure 9-3). The bid price on rate variables should be such that the win-rate on the low rate traffic is low enough that the sum of traffic matches the required KPI rate. We can define the formula below:

$$b_i{}^k = \begin{cases} b_i{}^P \cdot v_i{}^{k*}/V_t{}^k, \text{if } v_i{}^k \geq V_t{}^k \\ b_i{}^P \cdot \mathrm{s} \cdot v_i{}^{k*}/V_t{}^k, \text{otherwise} \end{cases} \quad (9.1)$$

This is the same formula as (9), but with a throttle added for low rate traffic. Appendix A shows a method of calculating a positive valued *s*, by recording the distribution of rates observed.

5.2 Error Minimization Calculation

We now have K bid prices $b_i{}^k$ that are each optimal for their particular KPIs. However we need to create a final bid price that behaves well for the multi-KPI problem; a replacement for equation (9.2). Let us define each KPI's disparity $\delta_i{}^k$ as a function of the difference between the constraint and the ideal value:

$$\delta_i{}^k = \left(\frac{V_t{}^k}{V_0{}^k}\right)^P \quad (14)$$

where $V_t{}^k$ is the current target for the KPI at time t and $V_0{}^k$ the original target entered by the advertiser, and P=2 is a power. This measures error in units of percentage difference from goal with an exponent to accentuate high misses. We now need to convert this measure of disparity into error that the system will try to minimize. Advertisers sign special contracts called Insertion Orders which specify the budget and required. Because of the necessity of meeting the contract, advertisers tend to view meeting the goal as important, over-performing as desirable but not essential, and missing as bad. Thus error is asymmetric, with over-performance much less important than achieving the minimum desired KPI goals. In order to capture this, we modify the raw KPI disparities to create a KPI error measure $\Delta_i{}^k$ that has a step discontinuity in which it de-weights disparities that are above goal by a factor r

$$\Delta_i{}^k = \begin{cases} \delta_i{}^k, if\ \delta_i{}^k > 1 \\ \left(r \cdot \delta_i{}^k\right), otherwise \end{cases} \quad (15)$$

$$r = \frac{\min(\mathrm{u}^k)\left(\frac{1}{w}-1\right)}{\sum_k \mathrm{u}^k - \min(\mathrm{u}^k)};\ w = 0.5 \quad (16)$$

Setting r as above ensures that at least 50% of the error will be allocated to KPIs that are failing their goals. Thus the system won't simply over-invest in one of the KPIs, at the expense of KPIs that are in trouble. In addition the scale-down in error upon reaching a target ensures that the system will continue to attempt to improve upon the solution, even if all KPIs are exceeding their targets.

We can now perform gradient descent on the error function (15). The solution with the greatest error reduction will be to select the bid price for the KPI that is most in need

$$b_i{}^* = b_i{}^k : \max \mathrm{u}^k \cdot \Delta_i{}^k \quad (17)$$

Successively minimizing the maximum error component is an example of *Tchebycheff Optimization* [12, 20, 26]. This form of optimization formally meets the criteria for weak Pareto optimality as it will converge to a solution that dominates other solutions, although as with other non-linear optimization strategies, the solution could become stranded at a local optimum [22]. We vectorize or "batch update" by taking a step in the direction of each sub-optimal bid price $b_i{}^k$, weighted by the magnitude of its error $\mathrm{u}^k \cdot \Delta_i{}^k$.

$$b_i{}^* = \frac{1}{\sum_k \mathrm{u}^k \cdot \Delta^k} \sum_{k \neq P}^{K} \mathrm{u}^k \cdot \Delta_i{}^k \cdot b_i{}^k \quad (18)$$

We'll refer to the above Multi-KPI bid update algorithm as "Px".

6 SOME THEORETICAL REASONS WHY ERROR MIN MAY WORK "BETTER" THAN A HARD CONSTRAINT APPROACH

In order to understand why the above optimization scheme may be preferable in practice than a constrained alternative, we will need to identify the distributions that are common in online advertising. There are two key functions: (i) the advertising response function and (ii) the price-value function.

Definition 1: Advertising Response. Advertising Response is typically defined as the advertising effect per unit of investment – either impressions or dollars [24]. In our problem, the advertising response function will be defined as the volume of KPI events generated for a given bid price at a given time:

$$V^k(b_i{}^*) = W(b_i{}^*) \cdot v_i{}^k(b_i{}^*) \cdot I_t(b_i{}^*) \quad (19)$$

Advertising Response has been studied experimentally for well over 50 years. Meta-studies of hundreds of experiments have concluded that the shape of this function is almost always convex, which means diminishing returns with higher advertising investment [18,19,20,29]. Taylor et. al. (2009) conclude that "the weight of empirical findings makes it possible to generalize that the advertising-response function is convex" [28]. In addition to empirical studies, advertising response can be shown to be convex under a simple assumption about a greedy optimization:

Lemma 1: Convexity under Greedy Optimization

A greedy advertising optimization process will buy the highest value per dollar inventory in order descending $\frac{V^k(b_i{}^*)}{b_i{}^*}$ until it reaches its budget [10, 13]. Because of the sort ordering, each subsequent unit of inventory being purchased will be the same or lower value per dollar. Therefore we can postulate that the following will be true under greedy optimization:

$$\frac{V^k(c \cdot b_i{}^*)}{c \cdot b_i{}^*} \leq \frac{V^k(b_i{}^*)}{b_i{}^*}; \text{ where } c \geq 1;\ \frac{V^k(c \cdot b_i{}^*)}{c} \leq V^k(b_i{}^*)$$

$$V^k(c \cdot b_i{}^*) \leq c \cdot V^k(b_i{}^*) \text{ where } c \geq 1 \quad (20)$$

We will use (20) in Lemma 4 and later to measure the relative cost of different solutions.

Definition 2: Price-Value Function

Let the price value function be the expected KPI value per impression at a given bid price $v^k(b_i{}^*)$. This is the "current auction price" for different levels of KPI quality.

Lemma 2: Price-Value Function is Linear in GSP auction
If the advertiser is paying b_i for the traffic, then they must make at least b_i from subsequent conversions in order to remain profitable. Thus the advertiser's Nash Equilibrium must be greater than or equal to their cost. Moreover, the "Single Item" Generalized Second Price (GSP) auction has a dominant strategy of bidding true valuation [4]. Therefore we would expect the advertiser's value to equal their bid price. Therefore we can postulate a linear relationship between KPI value per impression ${v_i}^k$ and bid price b_i, if indeed the KPI that the advertiser has specified, is a true unit of value for that advertiser.

$$ {v_i}^k(c \cdot b_i) = c \cdot {v_i}^k(b_i) \;\; ; {v_i}^k(c + b_i) = {v_i}^k(c) + {v_i}^k(b_i) \quad (21) $$

(21) will be used in Lemma 4 below to establish the relative value and KPI volume of different solutions. Although this only asserts linearity for an individual advertiser, we often observe linearity between bid price and KPI on real auctions (Figure 2), suggestive that indeed KPI is a measure of economic value. In addition, earlier work has suggested correlations between bid price and real world measures of value such as home price value [14]. We can now note several properties of the advertising problem:

Lemma 4: Number of KPI Events (a.k.a "Volume") declines as an Exponential Function of the Number of Constraints
Consider an unconstrained solution U with bid b_U , KPI v_U and spend B_U and a constrained solution X with KPI which is a factor of c higher, $v_X = c \cdot v_U$. Using (21) we note that because the KPI is higher, the bid price will also be higher $b_X = c \cdot b_U$. The amount spent under each solution must be the same $B_U = B_X$. By definition $I_X = \frac{B_X}{b_x} = \frac{B_U}{c \cdot b_U} = \frac{I_U}{c}$ (22); so with the higher bid price we have a reduction in inventory that can be purchased. (20) notes that buying higher priced inventory should result in diminishing returns in ability to win inventory; let us define $W(c \cdot {b_U}^*) = q \cdot c \cdot W({b_U}^*)$ where $q \in [0..1]$ (23). The volume of KPI events at the unconstrained solution will be $V_U = W({b_U}^*) \cdot I_U \cdot v_U$. The volume of events at constrained point will be equal to $V_X = c \cdot v^k({b_U}^*) \cdot W({b_U}^*) \cdot q \cdot I_U / c = q \cdot V_U$. If K constraints are now added, loss grows as a function q^K.

Lemma 5: Cost per KPI event (a.k.a "Efficiency") worsens as an Exponential Function of the Number of Constraints
Let Efficiency be defined as the Cost per KPI: $CPV = \frac{b_i}{W(b_i) \cdot {v_i}^k(b_i)}$
Growth in the denominator as a function of constraints is q^K, which means that CPV increases exponentially with constraints.

Lemma 6: Higher Volume under Non-Constraint Solution
This is a consequence of Lemma 4 since $V_X = q \cdot V_U; q \in [0..1]$

Lemma 7: Lower Total Error Inclusive of Delivery under Non-Constraint Solution
Let us define error as the squared difference between KPI and KPI-target, and budget-spent and target-spend. Error for unconstrained solution Err_U will be

$$ Err_U = \left({V_U}^k - {V_t}^k\right)^2 + (B_U - B_t)^2 $$

From Lemma 4 we know that the constrained solution will have $V_X = q \cdot V_U$ with the same budget $B_X = B_U$. Therefore we know that the error will equal:

$$ Err_X = Err_U + (1-q)^2 \cdot \left({V_U}^k - {V_t}^k\right)^2 $$

Since we know $q \in [0..1]$ then $Err_X \geq Err_U$.

7.1 Discussion

The intuition behind these results is that because of the shape of advertising distributions – linearity of price-value and convexity on volume per dollar - forcing an algorithm to squeeze the last few percent of performance by buying above a constraint, is likely to produce lower amounts of KPI for higher expense. The money would be better spent buying up the K other KPIs that are lower on their distributions, and so less costly.

In practice the loss of efficiency can be "eye watering": Table 3 shows that for identically set up ads, the last 1% of viewability, came at the expense of 50% of delivery. Most advertisers would consider that 1% of viewability a "rounding error" and would gladly forgo it to double their volume.

We have also introduced the concept of "error" which measures distance from the advertiser's target vector *inclusive of volume*. We suppose that advertisers have some intuitive concept of "distance from ideal", and that volume is part of that concept. The above lemmas show that given any constrained solution that is meeting KPIs, but not necessarily volume per the standard problem definition (2), a lower or same error solution exists which does not have the constraint restriction. The Px algorithm is designed to use gradient descent to search for that lower error solution.

Throughout the experiments next, we will compare the error and volume of each algorithm. We will show that, not only does this principle hold, but it results in a surprising amount of performance gain in real auctions. For example, as Table 3 suggests, doubling volume may be possible with a rounding error change in other KPIs. This kind of performance gain is significant in real-world advertising.

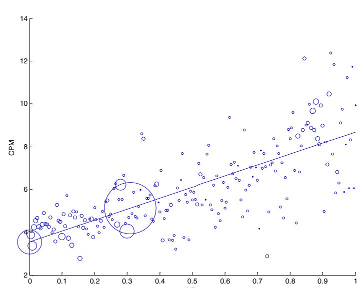

Figure 2: Viewability Rate of website versus Price paid by advertiser for One Video publishers on July 5, 2016. Bubble size represents the number of impressions from each website. $b_i({v_i}^k) = \beta_1 {v_i}^k + \beta_0$; $\beta_1 = 4.50$ ($t = 106; p < 0.01$); $\beta_0 = 5.47$ **(t=261; p<0.01).**

7 AD SERVING SIMULATION

We developed a Matlab auction simulation to illustrate the dynamics of each algorithm without the complexity of a real ad server [15]. In Section 10 we show how the simulation compares to actual ad-server experiments. The parameters were T=50 time periods, each with I_t = 300 impressions. The advertiser wishes to buy I=5000 impressions, and from these generate K=3500 KPI events, suggesting an initial KPI rate of ${V_0}^k = 0.70$. The advertiser has B=$50 dollars to spend; CPM B_0=10. T=50; I_t = 300; I=5000; K=3500; B=50; Z=5.

7.1 Ad Serving Loop

Each time period, following loop is executed:

1. Assume there are I_t impressions that are available to purchase at each time period t.

2. Set the predicted value of each impression ${v_i}^k$ to a uniform random number between 0 and 1. ${v_i}^k = \mathrm{U}(0,1)$

3. Estimate the pacing bid price ${b_i}^P$ as per Equation 8. Let $w = \sum_i {W_i}^* / \sum_i b_i$.. The pacing bid price ${b_i}^P$ can then be calculated as follows: At each time t the controller wishes to buy I_P impressions, which equals probability of win W_i multiplied by total impressions during the cycle I_t. Using the formula for W_i above we calculate ${b_i}^P$ as follows: $\mathrm{I}_P = W_i \cdot \mathrm{I}_t$; $\mathrm{I}_P = w \cdot {b_i}^P \cdot \mathrm{I}_t$; ${b_i}^P = \frac{\mathrm{I}_P}{(w \cdot \mathrm{I}_t)}$

4. The KPI bid price and final bid prices are then calculated using the control processes described earlier (Equation 18).

5. For each of the I_t impressions, the impression i is "won" W_i=1 if the bid multiplied by a uniform random number is greater than a threshold Z. ${W_i}^* = \mathrm{U}(0,1) \cdot {b_i}^* > \mathrm{Z}$

6. The actual value from the impressions is then set as ${v_i}^{k*}$

$${v_i}^{k*} = {v_i}^k + \rho^k;\ \rho^k = \mu^k \cdot \mathrm{U}(0,1) + \sigma^k \cdot N(0,1)$$

${v_i}^{k*}$ represents the actual value of the traffic and is equal to predicted value ${v_i}^k$ plus ρ^k and capped between 0 and 1 (not shown above). ρ^k is a term representing additive noise.

7. The budget and KPIs are then updated, and targets for budget and KPI calculated using the feedback control process. (Equation 10).

The simulation is initialized in a perturbed state where T_{INIT} time periods have already been completed with the system offset from its ideal target B_0 and ${V_0}^k$ by a perturbation of $\varepsilon_P B_0$ and $\varepsilon_k {V_0}^k$; $\varepsilon_P \in [0.5..1.5]$; $\varepsilon_k \in [0.5..1.5]$. The simulation then continues over the remaining time steps until time $T$and the final result in terms of Pacing and KPI performance are recorded.

7.1 Ad Server Simulation Results

Optimizer Algorithm 1: Pacing: Figure 3 shows the behavior of a Pacing algorithm (pacing bid only ${b_i}^* = {b_i}^P$) (Equation 8). This buys all traffic until it fills its budget. The resulting algorithm delivers fully ($B_T/B_0 = 1$). However since it just buys everything, the KPI result is poor; ${V_T}^k/{V_0}^k = 0.750$. The total KPI events delivered are equal to 1*0.750=0.750.

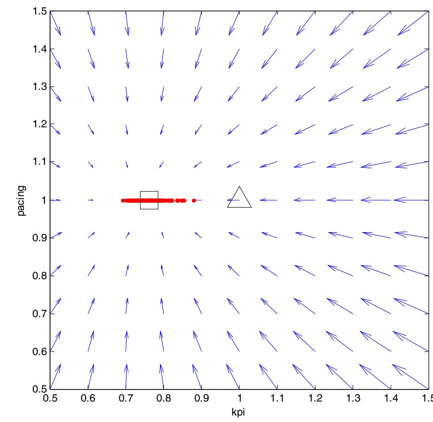

Figure 3: Pacing Optimization: y-axis is delivery achieved divided by delivery target (1.0 means impressions delivered were equal to the desired amount). x-axis is KPI achieved divided by KPI Target (1.0 means achieving the KPI target). Triangle represents the ideal (1,1) solution. The vectors show the trajectory of the control system from perturbed starting point to end state (dots). Square shows the mean for trajectory end-points. Pacing results in achievement of delivery goal (trajectories end at 1.0 on the y-axis), but poor KPI results (end-points are spread to the left below the KPI objetive; the end-point varies with severity of initial perturbation).

Optimizer Algorithm 2: Constraint: Figure 4 shows a single objective with constraints method (Equation 9). For each impression,if ${v_i}^k < {V_t}^k$ then ${b_i}^* = 0$, otherwise it is bid as per Equation 9. The constraints force good KPI performance (${V_T}^k/{V_0}^k = 1.1$). However with strict filter on traffic, the system now experiences delivery problems $B_T/B_0 = 0.750$. Total KPI events delivered is higher at 1.10*0.750=0.825.

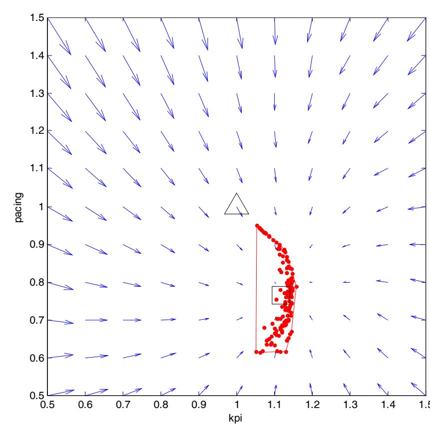

Figure 4: Constraint Optimization: Constraints push KPI results to be above target (right of the 1.0 vertical line), but result in problems pacing (below the 1.0 pacing horizontal line). Square indicates the mean of trajectory end-points. A convex hull surrounds the end-points.

Optimizer Algorithm 3: Px: Figure 5 shows the Error Minimizer algorithm (Equation 18). Px achieves very close to a perfect score with delivery and KPI just a fraction away from 1.0; $B_T/B_0 = 0.984$; ${V_T}^k/{V_0}^k = 0.995$. Total KPI events delivered equal 0.979. Thus, by allowing a tiny reduction in the KPI rate ($1 \rightarrow 0.995$), delivery was able to be increased in significantly (0.825 → 0.98). Intuitively this solution is a rounding error away from both KPI and delivery goals – so meeting *both* goals - and produces 20% higher KPI delivery than either pacing or constraint approaches.

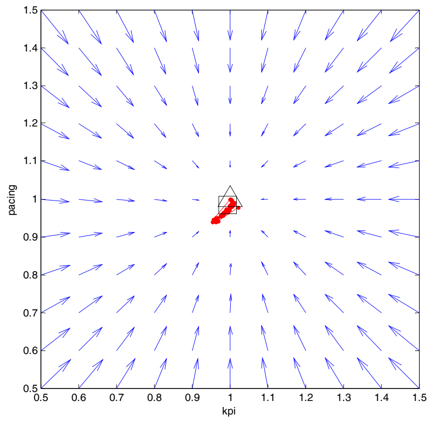

Figure 5: (top) Error Minimization: Error-min enables dynamic trading off between pacing and KPI. The error (distance between end point and (1,1)) is smaller than either pacing or constraint strategies. The square represents the mean of trajectory end-points. This is slightly shifted towards the lower left from the (1.0,1.0) ideal (the triangle shape is at 1,1). However for that tiny reduction in KPI and Pacing, the above solution produces 20% more events and much lower error.

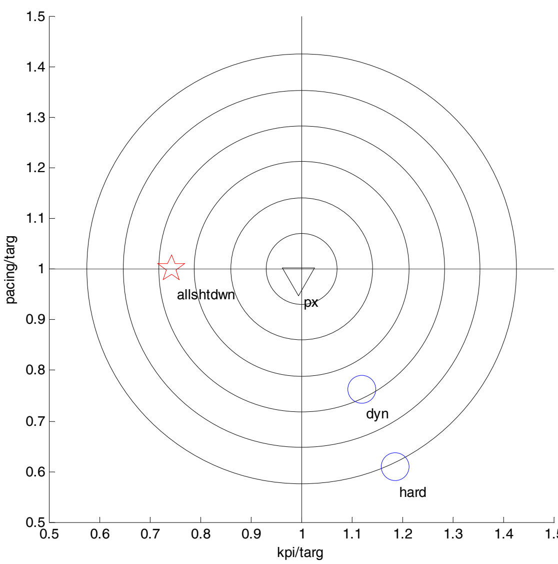

Figure 6: Mean final performance from multiple perturbation starts from different classes of algorithms from simulation (Table 1). Star is pacing, Circle constraint, and Triangle Px. x-axis is KPI achieved divided by KPI Target (1.0 means achieving the KPI target). y-axis is Pacing achieved divided by Pacing Target (1.0 means impressions delivered were equal to the desired amount). Px is close to center (a tiny amount left and down), pacing achieves delivery targets but has high error on KPI, and constraint methods exceed their KPI goals, but show a large miss on delivery.

Table 1: Simulation results

Simulation	Value[1]		
Metric	Pac μ	Con μ	Px μ
Ads	121	121	121
ImpsTarg	5,000	5,000	5,000
ImpsActual	5,000	3,817	4,922
ViewsTarg	3,500	3,500	3,500
ViewsActual	2,601	3,914	3,481
Views/Targ	0.743	0.854	0.979
RMSE	0.182	0.187	0.012
Imps/Targ	1.000	0.763	0.984
VR/Targ	0.743	1.118	0.995
VRTarg	70.0%	70.0%	70.0%
VRActual	52.0%	78.3%	69.7%

[1]Pacing = "Pac", Constraint = "Con", Error Min = "Px". *indicates worse than Px at p<0.05 level under t-test; + indicates better than Px at p<0.05 level under t-test.

8 AD SERVER EXPERIMENTS

We next examined the algorithms in a real ad-server. One Video is responsible for serving about 13.2% of all US video ads (Comscore, 2014). The code was implemented in ANSI C and included an A/B test switch to run ads on different algorithms.

In our first experiment, 35 ads were configured with the three algorithms introduced earlier – Pacing, Constraint, and Px. Each ad was set up with the identical parameters: I=5000; B=\$50; $b_i \leq \$12$; $V \in [3500,4000,4500]$; $V_0 \in [0.7, 0.8, 0.9]$. \$1,750 of real dollars were used to buy ads (\$50 per ad), and 5,000 impressions were set as the delivery target for each ad. The ads ran over a 30 day period. The KPI being targeted was Viewability Rate.

The above parameters were nearly identical to those used in the simulation. The results are shown in Table 2. Px took a tiny reduction on KPI compared to Constraints (VR/Targ = 0.997 versus 0.978; not significant on t-test). However delivery was significantly better for Px (0.70 → 0.98; p<0.05). Ignoring the variance, on average, for a 0.1% reduction in KPI, delivery was increased 1.4 times. Thus Px approximately achieved both KPI and delivery goals (0.997 and 0.98); but with 40% more delivery (Table 2, Figure 7; Figure 8).

8.1 Comparison to Simulation

We can compare the 35 production ads (Figure 7; Table 2) to the simulation results (Figure 6, Table 1) presented earlier. Lemma 7 suggested that non-constraint methods are able to generate lower error than constraint methods. Both simulation and production ads do indeed show lower RMSE for Px than both Constraint and Pacing solutions.

One difference between the simulation and production results is that the simulation suggested that constraint methods may over-perform their KPI targets; i.e. the Constraint position is slightly towards 5 o'clock, indicating KPI/targ results above 1.0. In contrast, in production we did not find much over-performance – Constraint produced an average of 0.997 (slightly less than 1.0), and only a few points exceeded the KPI target. We believe this may be due to regression-to-the-mean effects [27]. The act of buying high KPI traffic can be considered a form of "biased sampling". When sampling a population's high extremes, the errors are not symmetrically distributed – the actual "true mean" for the group will tend to lay further towards the population mean – i.e. lower than the predicted value. The magnitude of the correction towards the mean increases with (a) prediction error or lack of correlation of predicted values upon repeated testing and (b) the extremeness of the biased sampling. Therefore, we should expect a relatively high downward correction under real-world ad-serving conditions. If constraints are therefore being used because of the desire for the ad-server to "guarantee" that it will meet the advertiser's KPI, in practice actual KPI performance may very well still fail because of regression-to-the-mean. Advertisers should be aware of this when pursuing a constraint-based approach.

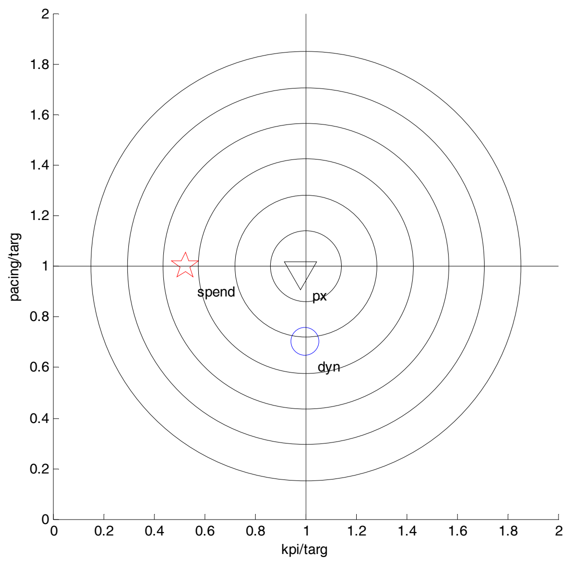

Figure 7: Means for 35 Production Ads under parameters similar to simulation (compare to Simulation in Figure 6). Star symbol indicates the results from Pacing ads (5 points). Inverted triangle shows results from Px ads (19), and Circles indicate results from constraint ads.

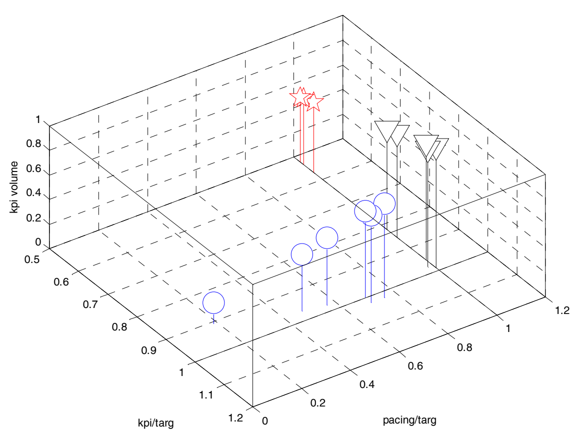

Figure 8: Subset of ads from the 35 ad experiment (Table 2) with x-axis KPI, y-axis Pacing, and z-axis representing total KPI events. Circles are constraint algorithms and Stars are Pacers. Px (triangles) produce far more KPI events than the other algorithms.

Table 2: Live Results from 35 ads

35 ads	Value					
Metric[1]	Pac μ	Con μ	Px μ	Pac σ	Conσ	Px σ
Ads	5	11	19			
ImpsTarg	25,000	55,000	95,000			
ImpsActual	25,010	38,656	93,291			
ViewsTarg	19,500	47,000	75,500			
ViewsActual	10,313	32,838	72,091			
Views/Targ	0.522*	0.706*	0.960	0.255	0.378	0.063
RMSE	0.338*	0.227*	0.057	0.180	0.205	0.045
Imps/Targ	1.000	0.703*	0.982	0.000	0.309	0.043
VR/Targ	0.522*	0.997	0.978	0.254	0.070	0.090
VRTarg	78.00%	85.45%$^+$	79.47%	4.5%	5.2%	6.2%
VRActual	41.23%*	84.96%$^+$	77.31%	21.0%	3.3%	4.9%
eCPM	9.46	7.29$^+$	9.51	0.88	2.86	3.49
vCPM	22.94*	8.58$^+$	12.30	20.85	3.44	4.55

[1] Pacing = "Pac", Constraint = "Con", Error-Min = "Px"; * indicates worse than Px at p<0.05 level under t-test; $^+$ indicates better than Px at p<0.05 level under t-test.

Table 3: Three ads with comparable goals

Ad[1]	VR	Imps	VR/ targ	Imps / targ	eCPM	vCPM
PAC – ad TT	48.0%	166	0.5333	1.000	6.13	12.77
CON – ad U	88.2%	94	0.980	0.566	6.06	6.87
PX – ad S	86.0%	166	0.956	1.000	5.24	6.09

[1] Three of the 35 ads for comparison with roughly the same targets: Impressions = 166 per day and ViewabilityRate= 90%. Px gets to 86% Viewability Rate. Constraint achieves 2% higher Viewability Rate of 88%. However Constraint *halves* the number of impressions to get this extra 2%. Constraints tend to promote economically irrational behavior that makes it hard to improve other KPIs and delivery.

9 RESULTS OVER 6 MONTHS

In April 2016 the new optimization functionality was made available to about 1/3rd of One Video's advertiser population. Between April and the end of September about 130 ads went live using the new optimizer, and 250 ads were initialized using a priority-based, constraint optimizer. Although the selection to each cohort was not randomized since advertisers opted in on their own, the results match the simulation and earlier production results.

The target viewability goals specified by the two groups of advertisers were not statistically significantly different (63.5% vs 61%(ns)), suggesting that both groups had similar targets. Yet Viewability rate delivered versus desired was significantly higher in the error-optimized group: 1.07 versus 0.64. There was minimal difference in delivery in this case (0.82 versus 0.80). Therefore the Px group experienced a 1.7x increase in KPI volume (53%->88%) (Table 4).

Table 4: 400 ads over 6 months

6 months	Value			
Metric	Con μ	Px μ	Con σ	Px σ
Ads	274	126		
ImpsTarg	2,174,652,928	274,418,086		
ImpsActual	2,290,497,954	290,721,874		
ViewsTarg	1,546,292,689	152,244,234		
ViewsActual	236,438,173	126,524,237		
Views/Targ	0.532	0.882*	0.499	0.539
RMSE	0.448	0.364*	0.259	0.252
Imps/Targ	0.80	0.82	0.32	0.32
VR/Targ	0.64	1.07*	0.52	0.49
VRTarg	61.0%	63.5%	0.13	0.15
VRActual	38.7%	66.0%*	0.31	0.32
eCPM	8.95	11.90*	3.24	4.18
vCPM	23.12	18.03*	3,392	65
Pr(Success)	29.3%	60.5%*		

* indicates significantly different from Legacy at $p<0.05$ under t-test.

10 HIGHER NUMBERS OF KPIS

It was also possible to report on multiple KPIs as part of the 6 month experiment. Table 5 shows all KPI tuples selected. For example, "Pacing+VR+Demo+CR" shows results for advertisers who had targets for Viewability Rate (VR) and Demographics (Demo) and Completion Rate (CR).

These KPI combinations all have KPIs in different units, making comparions difficult. For instance, the mean Clickthrough rate (CTR) is around 0.10%, where-as the mean Completion Rate (CR) is around 60%. In order to report a single number for performance, we therefore report the *average KPI lift over the mean.* For example, if Viewability Rate (VR) mean was 0.33 and CR mean 0.60, then an advertiser targeting VR and CR who achieved 0.66 and 0.70 would have lift of (0.66/0.33 + 0.70/0.60)/2 = 1.58x.

In the treatment group, Advertisers with 2 KPIs averaged about 2.54x lift (1.41x legacy). 3 KPIs averaged 1.44x and 1.28x (1.01x and 0.96x legacy), and 4 KPIs averaged 1.09x. Px therefore achieved higher lift in all comparable cases. It is also worth observing that as more KPIs are selected, the system produces lower lift. This is consistent with Lemma 5.

Table 5: Multi-KPI Results from 400 ads

6 months	Lift = Mean(KPI/Mean(KPI))				Ads	
Multi KPI Tuple[1]	Con μ_L	Px μ_L	Con σ_L	Px σ_L	Con N	Px N
Pacing+VR	1.41	2.54	1.04	1.12	132	78
Pacing+VR+CR	1.01	1.44	0.50	0.47	45	30
Pacing+VR+Demo	0.96	1.28	0.55	0.39	81	11
Pacing+VR+Demo+CR		1.09		0.08	0	7
Pacing+VR+CTR	0.55		0.13		5	0
Pacing+VR+CR+CTR	1.26		0.59		11	0

[1] Multi-KPI Results from 400 ads over 6 months on Px versus Legacy algorithm ("Leg"). VR="Viewability Rate", CR="Completion Rate", CTR="Clickthrough Rate", Demo="Demographic In-Target Rate". *N*=number of ads with this KPI tuple as its target. Each cell shows average lift across the KPI tuple. Empty cells mean there were no ads with this configuration.

11 PREVIOUS WORK

Most work in online advertising has focused on single objective problems, particularly click and conversion maximization [5,11,13,14] and smooth budget delivery [1,17,23,31,32]. Geyik et. al. tackle multiple KPIs but do so by solving sequential single optimum problems [6]: they propose a solution of using "prioritized goals", where the advertiser specifies which key performance indicator they care about the most, and that is met first, and then if others can be met, they are met only after the first priority (this approach is referred to as the *Lexicographic method* in the multi-objective optimization literature [25]). By using a prioritized goal approach, this enables the optimization problem to be translated into a series of *single variable - single constraint* maximization problems that are applied in succession.

The problem with "prioritized goal satisfaction" is if the system is unable to achieve the first objective, then all other KPI goals could be extremely poor. For example, if the budget is treated as a top-priority KPI, then the system risks buying "junk traffic" and never actually delivering any performance.

The present paper builds upon the Cost-min algorithm which has been studied and implemented over a long period of time in computational advertising [11, 13, 14]. This is used for calculating a bid price that will achieve a given Cost Per Acquisition constraint. The algorithm also utilizes *scalarization* to create a single objective function from multiple criteria; another well-known approach in the optimization literature [12, 20, 26]. An important modification to this general technique is that the "scalarized" objective function uses an error measure which is sensitive to each variable either achieving or failing its own goals. This gives rise to the algorithm's behavior of focusing resources on KPIs that aren't meeting their goals, and de-resourcing KPIs that have met their goals. Further work could be done shaping the penalty function to better reflect advertiser judgments of meeting their goals, but even with the simplistic squared-error - discontinuity function presented here, the results have been surprisingly good.

12 ECONOMICS

The impact of switching to error minimization at One Video has been very positive. As shown in Table 3 and 4, we have observed significant gains in KPIs and delivery for advertisers using the method. More qualitatively, we have also seen fewer support tickets. A customer survey performed just before release and then 6 months after release, measured advertiser satisfaction across 9 criteria. The most significant increase was for *"Performance against KPIs"*.

The economic impact of better KPI and delivery fulfillment is also considerable. Whilst not reporting One Video's revenues, we can note that ComScore publicly reported that One Video served 2.5 billion video ads during the month of July 2016 [2], and TubeMogul reports that a 30 second video ads average a CPM of approximately $10 [3]. Thus a One Video sized ad-server would be expected to generate around $300 million in revenue per year. Increasing the KPIs by a factor *c*, means the advertiser could raise their budget by *c* and still maintain at least the same revenue divided cost [13]. A 1.7x gain in delivery therefore translates into $206 million in additional revenue.

13 CONCLUSION

This paper has reviewed the literature on ad servers and proposed an alternative framework which uses the concept of multi-objective error minimization. Multiple experiments show that error minimization solutions are significantly closer to the advertiser's target, and most notably, address the "lack of delivery" phenomena that we see under the standard, constraint-based problem formulation.

Since the 1990s, Ad Server design has been invariably formalized as a single objective maximization problem subject to constraints (including work by the present authors eg. [13]). While constraints are an important tool, and can be used in the formulation described in this paper, it seems increasingly likely that many KPIs in high demand by advertisers today, are really indicators for high conversion probability.

Multi-KPI error minimization strategies seem to offer significant benefits for these kinds of KPIs and problems. They enable potentially infeasible problems to be tackled soundly and enable economic trade-offs between far/near and at-target KPIs to be codified using continuous error functions, so that rational investment can take place (constraint methods effectively state that at no price would a KPI that is a fraction lower than the target ever be purchased; if value is linear with the KPI then that could be considered extremely irrational). KPI error minimization also enables the system to maintain its ability to produce volume.

We believe that more rich and varied KPIs are likely in the advertising space, and as a result, these techniques will become more important in the future.

APPENDIX

Let $D(v)$ be a distribution of KPI values observed so far and $W(b)$ be a win rate model. Assuming accurate predictions $v_i{}^{k*}$ $=v_i{}^k$ (i.e. ignoring regression-to-the-mean effects), in equation 9.1 $s = 0$ will buy none of the below-rate traffic. This will trivially ensure that $\sum_t^T \sum_i^{I_t} W_i(b_i) \cdot v_i{}^k \geq V_t{}^k$, however this will also result in a KPI result that is overly high. We can buy a non-zero amount of the "below-rate" traffic by calculating $s \geq 0$ as follows:

$$s = \left(\mathrm{DL}\left(V_t{}^k\right) - \frac{V_t{}^k - \mathrm{VH}\left(V_t{}^k\right) \cdot \mathrm{DH}\left(V_t{}^k\right) - \mathrm{VL}\left(V_t{}^k\right) \cdot \mathrm{DL}\left(V_t{}^k\right)}{\mathrm{VH}\left(V_t{}^k\right) - \mathrm{VL}\left(V_t{}^k\right)}\right) / \mathrm{DL}\left(V_t{}^k\right)$$

$$\mathrm{VH}(V) = \frac{\sum_{v=V}^{1} v \cdot I(v)}{\sum_{v=V}^{1} I(v)};\ \mathrm{VL}(V) = \frac{\sum_{v=1}^{V} v \cdot I(v)}{\sum_{v=1}^{V} I(v)};\ \mathrm{DH}(V) = \frac{\sum_{v=V}^{1} I(v)}{\sum_{v=0}^{1} I(v)};$$

$$\mathrm{DL}(V) = \frac{\sum_{v=0}^{V} I(v)}{\sum_{v=0}^{1} I(v)};\ I(v) = W\left(b_i{}^P \cdot \frac{v}{V_t{}^k}\right) \cdot D(v);$$

ACKNOWLEDGMENTS

Thanks for Product support from Michael Chien (Moat release); Martina Ekman (Performance vs Delivery release), and Gary Yuen (Publisher release). Martina proposed 3 discretized weight configurations that were provided to advertisers with catchy names of "Performance", "Delivery" and "Equal" modes. Dmitry Perel and Abhishek Kumar were primary code reviewers. Thanks to Robert Luenberger, Nathan McCoy, Simon Asselin, David Roman and Seth Demsey for organizational support for R&D.

REFERENCES

[1] Chen, Y. and Berkhin, P., Anderson, B., Devanur, N. (2011), Real-Time Bidding Algorithms for Performance-Based Display Ad Allocation, KDD 2011, AUg 21-24, San Diego, CA.

[2] Comscore, (2016), Comscore VideoMetrix Video Ads Served by Company, accessed July 2016, https://www.comscore.com/Products/Audience-Analytics/Video-Metrix

[3] Crane, L. (2015), Global CPM Ranges, Tube Mogul Research, https://www.tubemogul.com/research/global-cpm-ranges/, accessed Jan 7, 2017.

[4] Easley, D. and Kleinberg, J. (2010), Networks, Crowds and Markets, Cambridge University Press.

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

[6] Geyik, S.C., Faleev, S., Shen, J., O'Donnell, S., Kolay, S., (2016), Joint Optimization of Multiple Performance Metrics in Online Video Advertising, Turn Inc., KDD 2016, San Francisco, CA.

[7] Gummadi, R., Key, P., Proutiere, A., (2011), Optimal bidding strategies in dynamic auctions with budget constraints, 49th Annual Allerton Conference on Communication, Control, and Computing, pp. 588-588. IEEE Publishers.

[8] IAB (2014), IAB Viewability Guidelines, Internet Advertising Bureau, http://www.iab.net/viewability

[9] Karlsson, N. (2011), Feedback Control and Optimization in Online Advertising, presentation, pp. 14.

[10] Karlsson, N. and Zhang, J. (2013), Applications of Feedback Control in Online Advertising. Proceedings of the 2013 American Control Conference, Washington, DC, June 17 - 19, 2013.

[11] Karlsson, N. (2016), Control Problems in Online Advertising and Benefits of Randomized Bidding Strategies, European Journal of Control, June 2016, pp. 31-49.

[12] Kaliszewski, I., (1987), A modified Tchebycheff metric for multiple objective programming. Comput. Oper. Res. 14, 315–323

[13] Kitts, B., LeBlanc, B. (2004), Optimal Bidding on Keyword Auctions, Electronic Markets - The International Journal of Electronic Commerce and Business Media, Vol. 14, No. 3.

[14] Kitts, B. Laxminarayan, P. and LeBlanc, B., Meech, R. (2005) A Formal Analysis of Search Auctions Including Predictions on Click Fraud and Bidding Tactics, ACM Conference on E-Commerce, Workshop on Sponsored Search, Vancouver, UK. June 2005.

[15] Kitts, B. (2016), Multi-KPI Controller Matlab simulation code, http://www.appliedaisystems.com/code/opt_simulation.zip, uploaded Jan 9, 2017

[16] Lee, K., B. Orten, A. Dasdan, and W. Li. (2012), Estimating conversion rate in display advertising from past performance data. In Proc. ACM KDD, pages 768–776.

[17] Lee, K., Jalali, A., Dasdan, A. (2013), Real time bid optimization with smooth budget delivery in online advertising, Proceedings of the Seventh International Workshop on Data Mining for Online Advertising. ACM, 2013

[18] Lodish, L., et. al. (1995a), How T.V. Advertising Works: A Meta-Analysis of 389 Real World Split Cable TV Advertising Experiments, Journal of Marketing Research, Vol. 32, No. 2, pp. 125-139.

[19] Lodish, L., et. al. (1995b), A Summary of Fifty-five In-Market Experimetns on the Long-term Effect of TV Advertising, Marketing Science, Vol. 14, No. 3, pp. 133-140.

[20] Marler, R. and Arora, J. (2004), Survey of multi-objective optimization methods for engineering, Struct Multidisc Optim 26, pp. 369–395.

[21] McMahan, H. et al. (2013), Ad click prediction: a view from the trenches. In Proc. ACM KDD, pages 1222–1230.

[22] Miettinen, K., (1999), Nonlinear Multiobjective Optimization. Boston: Kluwer Academic Publishers

[23] Quantcast, (2016), Control Pacing Advertising Budgets, https://www.quantcast.com/blog/control-design-for-pacing-advertising-budgets/ accessed August 8, 2016.

[24] Simon, J., Arndt. J. (1980), The shape of the advertising response function. Journal of Advertising Research, Vol. 20, August, pp. 11–28.

[25] Stadler, W. (1988), Fundamentals of Multicriteria Optimization. in: Stadler,W. (ed.) Multicriteria Optimization in Engineering and in the Sciences, pp. 1–25. Plenum Press, NY.

[26] Steuer, R. and Choo, E. (1983), An interactive weighted Tchebycheff procedure for multiple objective programming. Math. Program. 26, 326–344.

[27] Stigler, S. (1999). Statistics on the Table, Harvard University Press.

[28] Taylor, J., Kennedy, R. and Sharp, B. (2009), Is Once Really Enough? Making Generalizations about Advertising's Convex Sales Response Function, Journal of Advertising Research, June, 2009. pp. 198-200.

[29] Vakratsas, Demetrios, Tim Ambler. 1999. How advertising works: What do we really know? Journal of Marketing, Vol. 63, January, pp. 26–43.

[30] Wilkens, C., Cavallo, R., Niazedeh, R. (2017), GSP — The Cinderella of Mechanism Design, WWW 2017.

[31] Xu, J., Lee, K., Li, W., Qi, H., and Lu, Q. (2015), Smart Pacing for Effective Online Ad Campaign Optimization, KDD'15, August 10-13, 2015, Sydney, NSW, Australia

[32] Zhang, W., Rong Y., Wang, J., Zhu T., Wang, X. (2016), Feedback Control of Real-Time Display Advertising, WSDM 2016, Feb 22-25, 2016, San Francisco, CA.

[33] Garg, N., Young, N. (2002), On-Line End-to-End Congestion Control, Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science.