# Budget Pacing for Targeted Online Advertisements at LinkedIn

Deepak Agarwal
LinkedIn
dagarwal@LinkedIn.com

Souvik Ghosh
LinkedIn
sghosh@LinkedIn.com

Kai Wei
LinkedIn
kwei@LinkedIn.com

Siyu You
LinkedIn
syou@LinkedIn.com

## ABSTRACT

Targeted online advertising is a prime source of revenue for many Internet companies. It is a common industry practice to use a generalized second price auction mechanism to rank advertisements at every opportunity of an impression. This greedy algorithm is suboptimal for both advertisers and publishers when advertisers have a finite budget. In a greedy mechanism high performing advertisers tend to drop out of the auction marketplace fast and that adversely affects both the advertiser experience and the publisher revenue. We describe a method for improving such ad serving systems by including a budget pacing component that serves ads by being aware of global supply patterns. Such a system is beneficial for both advertisers and publishers. We demonstrate the benefits of this component using experiments we conducted on advertising at LinkedIn.

## Categories and Subject Descriptors

H.1.0 [**Information Systems**]: Models and Principles—*General*

## Keywords

budget pacing; targeted online advertising; generalized second price auction

## 1. INTRODUCTION

Targeted advertising is a significant source of revenue for many Internet companies. Members visiting a website are served ad impressions. Members are described by their profile which may include their demographic attributes like their geographic location, age, gender etc. Advertisers may target a segment of users specified by a combination of attributes. Advertisers are required to set up *campaigns* by specifying creatives, a target member segment, a bid value and a daily

budget. As described in [7] and [18], many internet companies use a generalized second-price auction (GSP) mechanism to determine which ads to show. A GSP ad auction is one where at each opportunity of an impression, eligible campaigns are ranked based on bids and the winner pays the bid of the second highest bidder.

A second price auction mechanism used for individual auctions is a greedy algorithm. Ad campaigns participate in every eligible auction until their daily budget is used up. Here by eligible auction for a campaign, we mean an opportunity of an impression to a member who belongs to the targeted member segment of the campaign. When a campaign's budget is used up it drops out of the auction marketplace for the rest of the day. The second price auction mechanism is greedy as it tries to maximize revenue at each opportunity independent of other auctions. This greedy mechanism is not optimal and has several practical drawbacks.

- Campaigns that perform well often use up their budget fast and drop out of the marketplace early in the day.

- Advertisers often get a skewed representation from members within the audience segment they target.

- As campaigns drop out, the competition in the marketplace reduces. In a second price auction mechanism, reduced competition results in reduced revenue.

- The ad serving infrastructure may have a delay in the availability of latest spend information and campaigns spending budget fast may exceed its budget. This results in over delivery to campaigns which is essentially loss of monetizing opportunity.

In this paper we describe a *budget pacing* algorithm that improves the ad serving mechanism on each of the above mentioned issues. The budget pacing algorithm aims to spread out impressions to every campaign evenly over a budget day. The main idea is that we serve impressions to campaigns by being aware of global traffic patterns. For each campaign we obtain a forecast for the traffic pattern of eligible impressions during the day. Based on the forecasts we determine an allocation plan in which we allocate spending budget of the campaign proportional to the forecasted eligible traffic. At runtime, we monitor the spend of each ad campaign closely and for campaigns that spend faster than the allocation plan, we throttle them and do not allow them to participate in some auctions.

Since this algorithm slows down the spending rate of campaigns that spend faster than the plan, it increases the lifetime of campaigns and helps to mitigate the problems described above. Most importantly, this is a desired feature for many advertisers since they get a more diverse audience. This improves revenue metrics as campaigns stay alive for longer and provide higher support for other campaigns in the GSP auctions.

We describe the system we implemented and deployed at LinkedIn and the experiments we conducted that proved our claim. Targeted advertising is a significant source of revenue for LinkedIn, see [2]. Members visiting LinkedIn.com are served ad impressions and advertisers pay LinkedIn on a pay-per-view (CPM) or a pay-per-click (CPC) mode. LinkedIn has a very rich data about members and allows advertisers to target member segments by attributes like their geographic location, industry, company, education, professional skills etc., or their activity attributes on LinkedIn.com like following companies, group memberships etc. Advertisers may target a segment of users specified by any combination of such attributes. Advertisements are ranked using GSP on expected revenue from the ads where the winner pays an amount so that the expected revenue from the impression equals that of the second highest ad.

Our implementation has been deployed in two different advertising products at LinkedIn.

- *Direct Ads* is a self service advertising product where advertisers set up campaigns via a web based UI. The advertisements appear on many pages within the the LinkedIn.com domain, e.g., the top and the side panels on the home page, the profile page, the contacts page etc. Direct Ads only appears on the desktop environment.

- *Sponsored Status Update* or SSU is an advertising product that serves advertising content on the update stream of members. The update stream of a member contains content generated by the member's connections, the groups the member belongs to, the companies the member follows etc. SSUs are contents that are sponsored by companies to be promoted on the update stream of members who do not follow the company. Such content would otherwise appear naturally on the update stream of members who follow the company. Advertisers may set up campaigns via a web based UI or via interactions with sales agents. The update stream along with the SSUs appear on the desktop, mobile and the tablet environments.

In section 2 we describe the algorithm in details. In section 3 we discuss the engineering design of our implementation at LinkedIn and In section 4 we discuss the design and the results of the experiments we ran at LinkedIn. The Direct Ads and the SSU are very different in terms of their age and status. We observe beneficial but very different effects of our pacing algorithm on these two products

## 1.1 Literature

Targeted advertising bears a lot of similarity to the Ad-Words problem, also known as search advertising, where advertisers target online search keywords. There is a rich literature studying various aspects of the AdWords problem, e.g., the study of game theoretic properties of such auctions in [7], [18] and [4]; advertiser optimizations like budget optimization in [9] and selecting profitable keywords in [16]; revenue maximization for the search engine provider in [13], [10], [8] and [12].

The AdWords problem is a generalization of the online bipartite matching problem. As explained in [13], the greedy algorithm to serve ads for AdWords has a worst-case competitive ratio of 1/2. Goel and Mehta [10] though showed that the greedy algorithm has a competitive ratio of 1-1/e in the random permutation input model and in the i.i.d. input model. Mehta et.al. described a simple deterministic algorithm in [13] that achieves an optimal competitive ratio of 1-1/e. There are many other articles that study practical aspects and methods for revenue and/or relevance maximization for search ads, see e.g., [15] and [11].

Member demographics based targeted advertising is significantly different from the AdWords problem in many ways. The main difference is that advertisers in AdWords target at the very atomic target level which are search keywords-they are able to set different bids for each different keyword. In targeted advertising, such as the one in use at LinkedIn, advertisers are not able to target at the most atomic level. Advertisers set the same bid value for all members in the target audience. This adds significant complexity to the problem and makes it difficult to obtain theoretical guarantees.

Mehta et.al. [13], [6], and [1] suggest using a modified bid of campaigns in every ad auction. In an auction marketplace where the competition is low, the modified bids of campaigns exhausting their budget is likely to reduce to very low amounts. Often times though a reserve price as in [14] is used for auctions that sets a lower limit on what an advertiser can bid and pay. Having a reserve price restricts how low the bid amount can get and hence this technique might not help in increasing the life of campaigns. Instead of using a modified bid for auctions we use probabilistic throttling to filter out the high performing campaigns from some randomly chosen auctions.

Our main contribution is that we developed and deployed a simple novel algorithm that improves advertiser experience and publisher revenue for targeted advertising. We set up appropriate real-life experiments and show the benefits of our deployed algorithm.

## 2. THE PACING ALGORITHM

Suppose $\mathcal{S}$ is the set of all members. An ad campaign $i$ targets a subset $\mathcal{S}_i$ of $\mathcal{S}$, has a bid $b_i$ and a daily budget $d_i$. The day is discretized into $T$ equal length time windows and $s_{i,t}$, for $0 \le t < T$, denotes the cumulative spend of campaign $i$ from the beginning of the day till the start of window $t$. We obtain a forecast for volume of eligible impressions for campaign $i$ during each time window- $f_{i,t}$ denotes the cumulative number of eligible impressions for campaign $i$ till the start of window $t$. We allocate the amount of budget we would like to spend for a campaign during each time window proportional to the forecasted volume of eligible traffic for the campaign during that window. Hence, the amount allocated to be spend from the beginning of the day till the start of window $t$ is

$$a_{it} := \frac{f_{i,t}}{f_{i,T}} d_i, \tag{1}$$

where $f_{i,T}$ is the forecasted total volume of eligible impressions for campaign $i$ for the entire day. We apply a filter during each ad auction. For each eligible auction a campaign is allowed to participate with probability $p_{i,t}$, the pass through rate(PTR). The value of $p_{it}$ is updated at the start of each time window by

$$p_{i,t} = \begin{cases} p_{i,t-1} * (1 + r_t) \wedge 1 & \text{if } s_{i,t} \leq a_{i,t} \\ p_{i,t-1} * (1 - r_t) \vee 0 & \text{if } s_{i,t} > a_{i,t}. \end{cases} \quad (2)$$

Here $0 < r_t < 1$ is an adjustment rate.

For a campaign $i$, the allocation curve $a_{i,t}$ is fixed at the start of the day and we control the spend curve $s_{i,t}$ by modifying the PTR $p_{i,t}$. We aim to keep the spending curve $s_{i,t}$ close to the allocation curve $a_{i,t}$ for every campaign $i$.

Mehta et.al. [13] suggest using a modified bid of campaigns in every ad auction. They show that using $b_i^*$,

not good method:
$$b_i^* = b_i \psi(s_{i,t}/d_i) \qquad \text{where, } \psi(x) = 1 - e^{x-1}. \quad (3)$$

is an optimal choice. We have already discussed in section 1.1 why we use probabilistic throttling instead of bid modification. For probabilistic throttling, it would have been natural to use $p_{i,t}^* = \psi(s_{i,t}/d_i)$ as PTR. In a linear system this would result in same expected payoff as in [13]. The GSP auction mechanism is clearly not a linear system though. We still tried this choice of PTR. We observed an improvement in revenue but not a significant increase in campaign life. This may seem counterintuitive since $\psi(x) \to 0$ as $x \to 1$. This happens due to a combination of two reasons. Firstly, this form of PTR does not depend on the rate of budget spend and can therefore be slow to react. Secondly, system inefficiencies also reduce the effectiveness of this method. Based on this experience we developed the dynamic method of choosing PTR as described in (2) where the value at the current time window is set as a small perturbation of the value at the previous time window. We observed remarkably good results with this approach.

## 2.1 Implementation Details

### 2.1.1 Update Frequency

We update the PTR for each campaign at 1 minute intervals. This frequent updating makes the system agile and reach a steady state fast.

### 2.1.2 Forecasting

Forecasting the number of eligible impressions for campaigns plays an important role in our algorithm. It is used to determine the allocation curve for each campaign. Inaccurate forecasts may cause unnecessary throttling and may result in budget not being used up for some good campaigns.

We use a tool that we built at LinkedIn using the method described in [3]. Note that, in order to forecast the number of eligible impressions for campaign $i$, we need to forecast the number of impressions from the member segment $\mathcal{S}_i$, which is the target audience for campaign $i$. We maintain a large list of important member segments which we call *base profiles*. For each base profile we update the historical time series data of number of ad impression opportunities. We use an ensemble statistical time series model to fit the up-to-date time series data for each of these base profiles daily. We obtain forecasts for each targeted segment $\mathcal{S}_i$ using correlation models described in [3].

### 2.1.3 Adjustment Rate

We set the adjustment rate $r_t$ at a constant 10%. This naive choice of the adjustment rate is not only the easiest to implement, but is also a very robust one. A more theoretically sound algorithm should involve the derivative $\partial s_{i,t}/\partial t$, the rate of budget spend while computing the PTR; a campaign that spends budget very fast needs to throttled more whereas a campaign spending budget just a little bit faster than the allocation curve needs to be throttled less. We chose the naive form for two main reasons. First, an estimate for $\partial s_{i,t}/\partial t$ is likely to be noisy. The budget spend curve is not smooth, especially for cost-per-click campaigns for whom it changes only when there is a click which is significantly rare compared to the impressions. Second, we adjust the PTR frequently and even if we are not at the ideal PTR at any time point we can reach there fast.

adjust it every 1min, can reach ideal value fast

### 2.1.4 Slow Start

We always start with a PTR of 10% for all campaigns. We call this a *slow start*. This may be a low starting value for some campaigns and may be a high value for some other campaigns. The advantage of a low starting value is that it gives the system the time to learn about individual campaigns. On one hand, if a campaign is spending its budget too slow then it can open up fully with in 25 minutes. On the other hand if a campaign is spending too fast then having a 10% PTR to start with gives the system more time to reach the correct PTR. For such campaigns starting from a high PTR may result in using up the budget before the system learns that it needs to set a low PTR.

It is sensible to customize the starting value of the PTR for individual campaigns. We aim to do that in future developments where we plan to make the initial value of the PTR for a campaign, a function of the demand and the forecasted supply for the campaign. The demand for a campaign is the number of impressions required to spend the budget of the campaign and the supply is the total number of impressions available for the campaign. It is natural to set a low initial value of PTR for a campaign with a high demand-supply ratio.

### 2.1.5 Fast Finish

We use statistical models to forecast future supply of impressions for each campaign. The models typically have high accuracy but there are always statistical errors. As a result there may be situations where the algorithm continuously sets lower values of PTR for a campaign than what should have been set. We try to mitigate this issue by *fast finish*- we modify the allocation curve slightly so that the algorithm assumes there will not be any traffic during the last two hours of the day. This way the pacing algorithm attempts to exhaust budget within 22 hours. This in turn gives campaigns that have been wrongly throttled earlier in the day, more opportunities to use up their budget later.

## 3. ENGINEERING DESIGN

In LinkedIn's ad serving system, data on advertiser events and member events continuously flows through a tracking pipeline. Advertiser events may include creation of a new campaign or a modification to an existing one, like change in bid or daily budget etc. User events include member page views that resulted in ad request or opportunity, ad
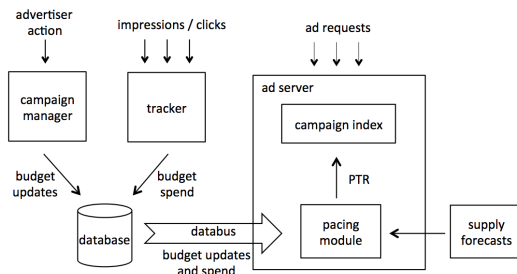
**Figure 1: Engineering design**

impressions served and user actions like click or share etc. This data is stored into a database.

The ad server maintains an index of all campaigns along with their current status. A pacing module that implements of our algorithm is deployed in the ad server. The pacing module receives this data from the database using a technology called databus, see [5]. Periodically the pacing module recalculates the PTR for each campaign using the latest budget/spend data and the allocation curve, and pushes it into the campaign index. The campaign index is scanned at ad matching and auction time, during which the PTR is used in a simple probability test to determine whether a given campaign should participate in the auction or not.

The forecast for eligible supply of impressions for campaigns is computed in an offline Hadoop system [17]. The forecast information is pushed to the pacing module on a daily basis.

As mentioned before, we update the PTR for each campaign at 1 minute intervals on average. The updates are randomly distributed across campaigns within each minute. That is, instead of updating all campaigns at once every minute, we update roughly 12% of all campaigns every 7 seconds. This updating frequency allows the system to reach a steady state fast and evenly spreads out the server load.

## 4. EXPERIMENTS

### 4.1 Experimental design

We designed and ran online experiments on the advertising products at LinkedIn to ascertain the benefits of deploying the budget pacing algorithm.

Experimental design, although an age-old topic of research, has been revolutionized by the Internet industry. Internet companies use experiments, often called A/B tests, to validate practically every step they take. In a typical A/B test, in order to compare control A with a treatment B, members are randomly split into two groups with one group receiving the control and the other group receiving the treatment. This approach is not directly applicable in what we wanted to test.

Randomly splitting traffic into two groups is not enough since the pacing algorithm would affect the budget spending for a campaign for an entire day and hence we would need to apply the same treatment to a campaign for the entire day. We could randomly split all campaigns into two groups and run different algorithms on two groups; this would help us determine the benefits of pacing to advertisers (like lifetime of campaigns etc.) but not the revenue benefits to

the publisher since campaigns from both the groups would participate in the same auctions.

In order to observe the full benefits of the algorithm it needs to be turned on for all campaigns for an entire day. We designed the experiment to run pacing on alternate days. By running the experiment for at least 2 weeks we are able to estimate the benefits of the pacing algorithm and control for any day specific bias and weekly seasonality that may be present in the system. Table 1 describes the experimental design.

**Table 1: Design of the experiment**

|        | Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|--------|-----|-----|-----|-----|-----|-----|-----|
| Week 0 | Off | On  | Off | On  | Off | On  | Off |
| Week 1 | On  | Off | On  | Off | On  | Off | On  |
| Week 2 | Off | On  | Off | On  | Off | On  | Off |
| ⋮      |     |     |     |     |     |     |     |

In our experiments we also controlled for any bias that may result due to trend. A weekly trend is of much smaller magnitude compared to the effect of our algorithm but by controlling for it we expect to get more accurate estimates.

When running the experiment for two weeks we considered 15 days of data for computing the metrics of interest. Starting the experiment from Sunday in Week 0 we continued till Sunday in week 2. The effect of pacing is measured by averaging over the 7 days when the algorithm was turned on during this period. The metrics corresponding to the control was measured using data from 8 days where we take an average of Sunday in week 0 and Sunday in week 2 to estimate the Sunday effect. Every other day gets an equal weight of 1/7. This mechanism helps to eliminate any linear trend present in the system.

Another way to run an experiment to determine the effects of pacing would be one where we create two independent ad serving systems and put in all the ad campaigns in both the systems with half the daily budget. We can then split incoming traffic and randomly allocate to the two different systems. The advantage of this design would be that we would not have to be concerned about weekly seasonality or trend or other daily variations. We chose not to use this design for two reasons. First, It would be difficult to apply the algorithm to campaigns with small daily budget and second, it would take a lot of time and effort to implement such a two-fold ad serving system which is likely not cost effective.

### 4.2 Metrics of importance

The following are the main metrics we measure in our experiments. For most metrics we take the case when pacing algorithm is turned off as baseline and report the percentage change when the algorithm is turned on. The metrics are computed using all the data from LinkedIn without any sampling. LinkedIn gets tens of millions of page views regularly and has tens of thousands of advertisers advertising at LinkedIn on any day. Given the volume of the data we are confident about the statistical significance of the metrics.

- Advertiser centric metrics

  - *Campaign life time*: We use the time taken to exhaust 95% of the budget for a campaign to mea-

sure the life of the campaign in a budget day. We report the median life-time of campaigns in hours.

- *Unique impressions per spend* is the ratio of number of unique members viewing an ad over the budget spend for the ad. An increase in this metric denotes an increase in the audience reach for advertisers with the same advertising spend.

- *Number of campaigns served* measures the total number of ad campaigns that were served on a day.

- Publisher centric metrics

  - *Cost per request*: Ratio of total revenue with total number of requests or opportunities is measure of revenue per unit opportunity.

  - *Over delivery*: The dollar amount we delivered to a campaign beyond its daily budget as a percentage of total revenue.

- Member centric metrics

  - *Unique campaings served*: We measure the average number of unique ad campaigns viewed by members. This is a measure of diversity of advertisement content viewed by members and an increase in this metric is an improvement in member experience.

## 4.3 Direct Ads

Direct Ads is the advertising product that serve advertising content on various LinkedIn webpages. This product is more than 4 years old and the demand from advertisers exceed the supply. There is a high coverage in most member segments.

We ran the experiment in Direct Ads for 2 weeks during 2013 Q4. The key results are tabulated in table 2. As we mentioned before, due to privacy issues we report most metrics in relative form with the baseline being the case when the pacing algorithm in not running.

### Table 2: Results for Direct Ads

| Pacing | Off | On |
|---|---|---|
| Campaign life time (hours) | 13.54 | 19.50 |
| Unique impressions per spend | - | +7.74% |
| Number of campaigns served | - | +4.74% |
| Cost per request | - | +5.67% |
| Over delivery | 3.8% | 3.4% |
| Unique campings served | - | +0.15% |

The metrics reveal that the pacing algorithm provides benefits to both advertisers and LinkedIn. The algorithm increases campaign life time by 44%. Advertisers are able to reach to a broader audience with the same budget and more advertisers are served impressions.

There is a benefit to LinkedIn as the cost per ad request increases. The pacing algorithm keeps campaigns alive for a longer time in a budget day. As discussed before, this helps increase competition later in the budget day. This is reflected in figure 2 which plots the variation in cost per click during a budget day. Cost per click (CPC) denotes what advertisers pay on average per click and in a second price

auction it is a measure of support price for auctions. It is primarily used for campaigns who pay per click and they are the dominant kind in Direct Ads. In figure 2, the solid red line denotes the average CPC curve during each hour within a budget day on the days when pacing was turned on. The blue dotted line denotes the same for the days when pacing was turned off. We see that with pacing turned on, the CPC is lower during the initial part of the day but higher for a much longer period during the second half of the day. The CPC is lower at the start of the day because we throttle all the high performing campaigns during that time who otherwise all compete together and exhaust their budget fast. Again, the plot was created from all the data collected by LinkedIn during the two weeks of experimentation. Figure 3 shows the variation in cost per request during a budget day.

The loss due to over delivery of impressions to campaigns is reduced by 10%. There is a little time delay in the availability of the latest budget spent information in LinkedIn's ad serving system. In absence of pacing, the good campaigns spend at a fast rate and because of the delay in latest budget information, the system often serves impressions to such campaigns even after the budget for the day has been spent. The pacing algorithm slows down the rate of spend of such campaigns and reduces the chances of over delivery.

We do not see a statistically significant effect of pacing algorithm on the diversity of ad content seen by members. That is likely because of the long tail of the distribution of the number of unique campaigns seen by a member.
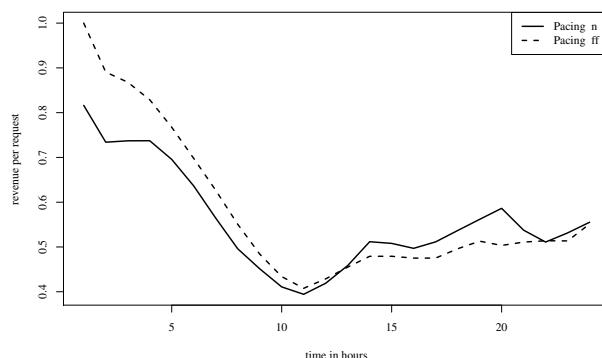


**Figure 2: Variation in average cost per click during a day in Direct Ads (scaled so that the baseline at time 0 is set to 1)**

## 4.4 Sponsored Status Updates

Sponsored Status Updates is a new advertising product from LinkedIn that was launched in mid 2013. Advertisers can sponsor a 'status update' and promote it to a specified segment of LinkedIn members. The sponsored content from an advertiser is blended with the organic content in the update stream and delivered to the members in the targeted audience.

The Sponsored Status Update marketplace was very different from Direct Ads at the time we ran the experiment. SSU is a new product, the demand to supply ratio is much lower but is growing rapidly. The SSUs appear on a prime
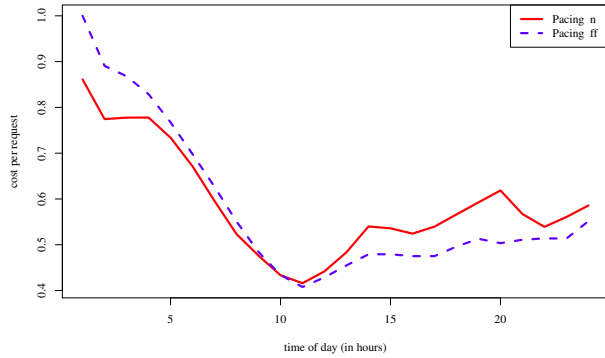
**Figure 3: Variation in average revenue per request during a day in Direct Ads (scaled so that the baseline at time 0 is set to 1)**

space in the LinkedIn webpage and smart phone/tablet apps. It has a higher visibility because of which campaigns can potentially exhaust the daily budget very fast. Pacing is thus very important for SSU. We ran a two week experiment on this product and the key results are tabulated in table 3.

**Table 3: Results for Sponsored Status Updates**

| Pacing | Off | On |
|---|---|---|
| Campaign life time (hours) | 6.92 | 17.25 |
| Unique impressions per spend | - | +10.52% |
| Number of campaigns served | - | +0.25% |
| Cost per request | - | +0.96% |
| Over delivery | 4.12% | 2.39% |
| Unique campings served | - | -0.04% |

It is expected that the pacing algorithm will have a much different effect in SSU and that is what we observe. First, we see that the increase in cost per request is of much lower magnitude compared to that in Direct Ads. The reason for that is the total revenue for SSU (at the time of the experiment) is largely determined by the advertiser demand which is lower than the supply. Second, the increase of campaign median life time is much more drastic (149%) compared to what we observed in Direct Ads.

We observe an improvement in the audience reach for advertisers as the average number of unique impressions per spend increased by 10.52%. There is a small increase in the average number of campaigns that got served per day, but this change is not statistically significant. We also observe a marked decrease in loss due to over delivery (42%). The effect on the average number of unique campaigns seen by members was slightly lower but this change is statistically insignificant.

## 4.5 Some Examples

Here we show the behavior of two specific campaigns during our experiments. Albeit two examples, in conjunction with the results described in sections 4.3 and 4.4, they further illustrate how well the pacing algorithm performs.

Figure 4 shows the performance of a specific campaign in SSU in two consecutive week days during the experiment.

Pacing was turned off on day 1 (blue dotted line) and was running on day 2 (red solid line). On day 1 the budget was spent with in the first 30 mins of the day and there was more than 20% loss due to over delivery. On day 2, the campaign was active till the 20th hour of the day and had a significantly lower over delivery loss.
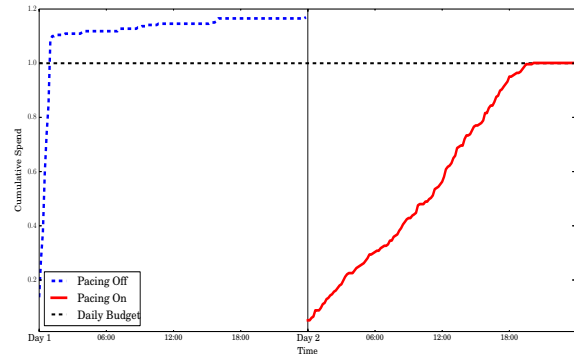


**Figure 4: Cumulative budget spent for a campaign in SSU (scaled so that the total budget is 1)**

Figure 5 demonstrates for a specific campaign that we are able to align the actual spending curve with the allocation curve well. The blue line is the observed spending curve and the red dotted line is the allocation curve for the campaign. This behavior is observed consistently for most campaigns with a reasonably high daily budget.
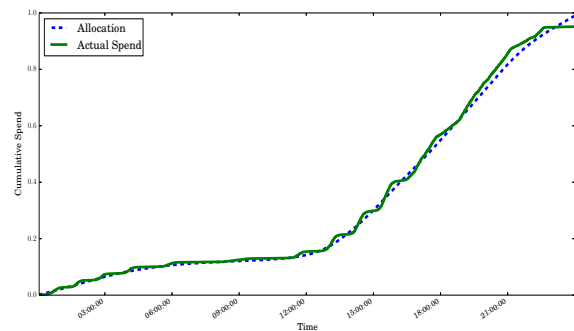


**Figure 5: Actual spend curve vs. the allocation curve for a campaign in SSU(scaled so that the total budget is 1)**

## 5. CONCLUSION

We develop a budget pacing algorithm that we use to evenly distribute the spend of advertising campaigns in a day. We implemented the algorithm and it is deployed in two advertising products at LinkedIn. The algorithm helps improve advertiser experience and provides revenue benefits to LinkedIn at the same time. We ran real life experiments at LinkedIn to demonstrate the benefit of using this algorithm. Based on the results of the experiments this algo-

rithm was been fully deployed in both the advertising products at LinkedIn.

# 6. REFERENCES

[1] Z. Abrams, O. Mendelevitch, and J. Tomlin. Optimal delivery of sponsored search advertisements subject to budget constraints. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 272–278, New York, NY, USA, 2007. ACM.

[2] D. Agarwal. Computational advertising: The linkedin way. In *Proceedings of the 22Nd ACM International Conference on Conference on Information &#38; Knowledge Management*, CIKM '13, pages 1585–1586, New York, NY, USA, 2013. ACM.

[3] D. Agarwal, D. Chen, L.-j. Lin, J. Shanmugasundaram, and E. Vee. Forecasting high-dimensional data. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1003–1012, New York, NY, USA, 2010. ACM.

[4] G. Aggarwal, A. Goel, and R. Motwani. Truthful auctions for pricing search keywords. In *7th ACM Conference on Electronic Commerce*, pages 1–7. ACM, 2006.

[5] A. Auradkar, C. Botev, S. Das, D. De Maagd, et al. Data infrastructure at LinkedIn. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1370–1381, April 2012.

[6] C. Borgs, J. Chayes, N. Immorlica, K. Jain, O. Etesami, and M. Mahdian. Dynamics of bid optimization in online advertisement auctions. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 531–540, New York, NY, USA, 2007. ACM.

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

[8] J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126. IEEE, 2009.

[9] J. Feldman, S. Muthukrishnan, M. Pal, and C. Stein. Budget optimization in search-based advertising auctions. In *8th ACM Conference on Electronic Commerce*, pages 40–49. ACM, 2006.

[10] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 982–991. ACM-SIAM, 2008.

[11] S. Lahaie and D. M. Pennock. Revenue analysis of a family of ranking rules for keyword auctions. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, EC '07, pages 50–56, New York, NY, USA, 2007. ACM.

[12] S. Lahaie, D. M. Pennock, A. Saberi, and R. V. Vohra. *Algorithmic Game Theory, Chapter: Sponsored Search Auctions*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[13] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized on-line matching. *Journal of the ACM*, 54(5):Article no. 22, October 2007.

[14] M. Ostrovsky and M. Schwarz. Reserve prices in internet advertising auctions: A field experiment. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 59–60, New York, NY, USA, 2011. ACM.

[15] F. Radlinski, A. Broder, P. Ciccolo, E. Gabrilovich, V. Josifovski, and L. Riedel. Optimizing relevance and revenue in ad search: A query substitution approach. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 403–410, New York, NY, USA, 2008. ACM.

[16] P. Rusmevichientong and D. P. Williamson. An adaptive algorithm for selecting profitable keywords for search based advertising services. In *7th ACM Conference on Electronic Commerce*, pages 260–269. ACM, 2006.

[17] R. Sumbaly, J. Kreps, and S. Shah. The big data ecosystem at LinkedIn. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 1125–1134, New York, NY, USA, 2013. ACM.

[18] H. R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, December 2007.