Proceedings of Machine Learning Research 109:46–58, 2019 ORSUM 2019

Optimal Delivery with Budget Constraint in E-Commerce Advertising

Chao Wei, Weiru Zhang* WEICHAO.WC,WEIRU.ZWR@ALIBABA-INC.COM
Shengjie Sun SHENGJIE.SSJ@ALIBABA-INC.COM
Fei Li FAKER.LF@ALIBABA-INC.COM
Xiaonan Meng XIAONAN.MENGXN@ALIBABA-INC.COM
Yi Hu ERWIN.HUY@ALIBABA-INC.COM
Kuang-chih Lee KUANG-CHIH.LEE@ALIBABA-INC.COM
Hao Wang LONGRAN.WH@ALIBABA-INC.COM
Alibaba Group
Hangzhou, China

Editors: João Vinagre, Alípio Mário Jorge, Albert Bifet and Marie Al-Ghossein

Abstract

Online advertising in E-commerce platforms provides sellers an opportunity to achieve potential audiences with different target goals. Ad serving systems (like display and search advertising systems) that assign ads to pages should satisfy objectives such as plenty of audience for branding advertisers, clicks or conversions for performance-based advertisers, at the same time try to maximize overall revenue of the platform. In this paper, we propose an approach based on linear programming subjects to constraints in order to optimize the revenue and improve different performance goals simultaneously. We have validated our algorithm by implementing an offline simulation system in Alibaba E-commerce platform and running the auctions from online requests which takes system performance, ranking and pricing schemas into account. We have also compared our algorithm with related work, and the results show that our algorithm can effectively improve campaign performance and revenue of the platform.

Keywords: ad allocation, e-commerce advertising, simulation system

1. Introduction

E-commerce platforms such as Amazon, eBay and Alibaba run hundreds of millions of auctions to sell advertising opportunities. Online advertising plays a crucial role in connecting advertisers and audiences, and generates tremendous value for E-commerce platforms. There are different types of online advertising which includes performance-based advertising, branding guaranteed advertising. We focus on the performance-based advertising in the paper. In this marketplace, advertisers can specify the maximum daily amount they are willing to pay and get the audience through various pages in E-commerce platform. The objectives of performance-based advertisers are usually to spend out the budget to maximize the performance goals (e.g., clicks, conversions as many as possible). Meanwhile, the ad serving system is optimizing revenue on behalf of the platform.

1. Both authors contributed equally to this research.

One of the central issues for the ad serving system of E-commerce platform is matching ads to requests with these objectives above, which can be formulated as a constrained optimization problem. There are many challenges to achieve all the objectives simultaneously in a complex competition environment. Each individual campaign has its own budget and performance goal, and there are hundreds of thousands of campaigns which compete with each other to acquire inventory in the marketplace. These varieties make the optimization extremely difficult.

In this paper, we present our work on optimal delivery in E-commerce platform, which can be formulated as a constrained optimization problem that maximizes specified goals and subjects to budget constraints. Our contribution can be summarized as follows:

- We propose an approach based on linear programming to optimize overall revenue of the platform and improve different performance goals in campaign level simultaneously. Compared with previous work, our approach can be run over all traffic and implemented in both display and search advertising.

- We design a simulation system to evaluate the results of different allocation plans by replaying auction from online requests. The proposed approach has been implemented in conjunction with pricing and ranking schemes in Alibaba E-commerce platform, and the results show that it can effectively improve both revenue and various performance goals.

2. Related Work

Most existing work related to optimizing budget constrained spend focused on bid modification and allocation. Allocation treats bids as fixed, and allows only decisions about whether the advertiser should participate in the auctions while bid modification is in a setting where bids can be changed Karande et al. (2013). Bhalgat et al. (2012); Chen et al. (2011); Zhang et al. (2014); Devanur et al. (2011) formulated display ad allocation problem by combining budget and bid optimization. Zhu et al. (2009); Agarwal et al. (2014) used probabilistic throttling to achieve budget control in sponsored search advertising. Xu et al. (2015) suggested using allocation or throttling to directly influence budget spending while bid optimization changes the win-rate to control. We also choose allocation strategy for consideration that advertisers in E-commerce platform usually set fixed price or price range for each campaign so that the performance of bidding optimization is limited.

There are several papers close to our work. Abrams et al. (2007) solved the problem by an optimal algorithm, but it can only run over head queries. Our approach uses request instead of query in order to be implemented in both search and display advertising. Karande et al. (2013) used a non-optimal algorithm and thought it is fairer than optimal algorithm. We still use optimal algorithm because it can avoid unnecessary competitions and get better performance which is proved by the experiments in Section 5. Motivated by Chen et al. (2011) and Xu et al. (2015), we define a multi-objective optimization problem, and convert it into a single-objective optimization problem with constraints since there may be multiple Pareto optimal solutions to a multi-objective optimization problem. We try to maximize the overall revenue and improve different performance goals simultaneously based on linear programming. The widely noted work by Mehta et al. (2013) presented the original formulation and there are related work such as Zhang et al. (2018); Chen et al. (2012); Lee et al. (2013); He et al. (2013); Chervonenkis et al. (2013) extended this framework to solve specific allocation problems such as dynamic ad allocation, video-ad allocation and smooth budget delivery.

3. Problem And Formulation

3.1. Preliminaries

What most ad serving systems do is to let advertisers participate in auctions as many as possible until their budgets have been exhausted, and then make them ineligible for the rest of the day. Obviously, this strategy causes stronger auction competition at the beginning of the day and yields very biased traffic to the advertisers, which is hard to maximize advertisers' performance goals. In this paper, we study an online allocation strategy to achieve different performance goals and maximize revenue simultaneously. Before formulating this problem, we define some basic concepts. Thus we denote by:

When a user browse a page (in the recommender system) or types a keyword query (in the search engine), there is a request sent to the ad serving system. $\Re = \{r_1, r_2, ..., r_M\}$ is the request stream which arrives over the whole day.

O is the set of ads.

$c_k \in C = \{c_1, c_2, ..., c_N\}$ is a campaign comes with a daily budget $budget_k$. Advertisers use campaign as a minimum marketing unit to achieve their specific performance goal.

We consider auctions of each request, where there are bidders competing for P slots. $L_i = \{o_p : o_p \in O, p = 1, 2, ..., P_i\}$ is an ordered set of ad indices called *bidding landscape* , where P_i is the number of ads in the landscape, and o_p is recalled and ranked by the ad serving system. Each *bidding landscape* L_i can be mapped into a set of slates.

$L_i^j = \{o_l^j : o_l^j \in L_i, l = 1, 2, ..., P_i^j, P_i^j \leq P\}$ is slate j for request i, where P is the maximum number of slots available for advertising on the page. *slate* can be obtained by deleting ads of L_i (while maintaining the ordering) and then truncating (if necessary) to P_i^j (at most P_i) ads. Generally speaking, P is much smaller than P_i. *Slate* represents a unique subset of a *bidding landscape*.

These two crucial concepts of *bidding landscape* and *slate* of ads we define are proposed by Abrams et al. (2007).

Ω_k refers to the set of slates including campaign k.

x_{ij} is a binary variable indicating whether slate j is assigned to request i ($x_{ij} = 1$) or not ($x_{ij} = 0$).

3.2. Problem Definition

Given the above notations, we can formulate the problem mentioned before as a multi-objective linear programming (MOLP) problem as follows:

$$\begin{aligned}
\max \quad & \sum_{i,j} rev_{ij} \cdot x_{ij} \\
\max \quad & \sum_{k \in C_1} \sum_{i,j \in \Omega_k} ctr_{ijk} \cdot x_{ij} \\
\max \quad & \sum_{k \in C_2} \sum_{i,j \in \Omega_k} cvr_{ijk} \cdot x_{ij} \\
s.t. \quad & \forall k, \sum_{(i,j) \in \Omega_k} cost_{ijk} \cdot x_{ij} \leq budget_k \\
& \forall i, \sum_{j} x_{ij} \leq 1, \\
& \forall i,j, \quad x_{ij} \geq 0.
\end{aligned} \tag{1}$$

In Eq.(1), rev_{ij} denotes the expected revenue from slate j for request i. $cost_{ijk}$ is the expected cost of campaign k when slate j is assigned to request i. The first objective is the expected revenue of platform. In real scenarios, advertisers often prioritize their performance objectives for different campaigns. Other objectives of Eq.(1) are the different performance goals in campaign level such as click and conversion. As there usually exist multiple (possibly infinite) Pareto optimal solutions, it's very difficult to solve such a multi-objective optimization problem. Although the MLOP tries to maximize performance of each campaign, our model can not insure that each campaign gets better result theoretically. Accordingly, we combine all the performance goals of campaigns into several performance constraints and convert the original problem with multiple objectives into a single-objective optimization problem with the constraints. Finally, we formulate the problem as a single-objective linear programming (SOLP) problem with the constraints that advertisers bid for more clicks and conversions in campaign level which is suitable for E-commerce scenario. The formula is presented as follows:

$$\begin{aligned}
\max \quad & \sum_{i,j} rev_{ij} \cdot x_{ij} \\
s.t. \quad & \sum_{k \in C_1} \sum_{i,j \in \Omega_k} ctr_{ijk} \cdot x_{ij} \geq T_{cy} \\
& \sum_{k \in C_2} \sum_{i,j \in \Omega_k} cvr_{ijk} \cdot x_{ij} \geq T_{vy} \\
& \forall k, \sum_{(i,j) \in \Omega_k} cost_{ijk} \cdot x_{ij} \leq budget_k \\
& \forall i, \sum_{j} x_{ij} \leq 1, \\
& \forall i,j, \quad x_{ij} \geq 0.
\end{aligned} \tag{2}$$

ctr_{ijk} is the estimated click-through rate of ad from campaign k of appearing in slate j for request i while cvr_{ijk} is the estimated conversion rate. These CTRs and CVRs are predicted based on the real log data and will be used in our algorithm and the simulation system described in Section 4.3. C_1 and C_2 are the sets of campaigns that bid to achieve more clicks and conversions respectively. Thus the problem is defined by grouping campaigns with the same performance goal(ctr or cvr) and setting click constraint T_{cy} and conversion constraint T_{vy}. The constraint setting is discussed in Section 5.

The dual of (2) can be written as follows:

$$\begin{aligned}
\min \quad & \sum_k \alpha_k budget_k + \sum_i \beta_i - \gamma T_{cy} - \delta T_{vy} \\
s.t. \quad \forall i, j, \quad & \beta_i \geq rev_{ij} - \sum_{k \in \Omega_{(i,j)}} \alpha_k cost_{ijk} \\
& \qquad + \gamma \sum_{k \in C_1} ctr_{ijk} + \delta \sum_{k \in C_2} cvr_{ijk} \\
\forall i, \quad & \beta_i \geq 0 \\
\forall k, \quad & \alpha_k \geq 0 \\
& \gamma \geq 0 \\
& \delta \geq 0
\end{aligned} \tag{3}$$

The parameters α_k, β_i, γ and δ are the dual variables of constraints in Eq.(2). There are natural interpretations that α_k can be interpreted as the minimum revenue margin required by a campaign, and β_i is the shadow price of satisfying an additional request i with a slate. In a way, γ and δ represent the trade-off between revenue and advertisers' performance goals (click and conversion). α_k, γ and δ will be used in the online algorithm in Section 3.3. $\Omega_{(i,j)}$ is the set of campaigns in slate j for request i.

3.3. The Real-Time Algorithm

Inspired by the previous work of Chen et al. (2011); Zhang et al. (2018), we develop a real-time algorithm based on the complementary slackness theorem to solve the optimization problem with various constraints mentioned above. The algorithm is shown as follows.

For an incoming request i, the ad serving system yields the bidding landscape L_i by retrieving and ranking ads. Each slate j can be obtained by deleting ads of L_i (while maintaining the ordering) and then truncating (if necessary) to P_i^j ads. We will detail how to generate the slate candidates efficiently in next section. Then we calculate rev_{ij}, $cost_{ijk}$ and ctr_{ijk} for each (i,j,k) which will be introduced in Section 4.1. Finally we compare each slate j for request i with the score of $(rev_{ij} - \sum_{k \in \Omega_{(i,j)}} \alpha_k cost_{ijk} + \gamma \sum_{k \in C_1} ctr_{ijk} + \delta \sum_{k \in C_2} cvr_{ijk})$, and slate j with the maximal value will be chosen.

Chen et al. (2011) have proved that the online algorithm uses complementary slackness theorem to assign slate to request with the highest scores such that the offline optimality can be preserved. The input α, γ and δ can be calculated in advance with the real log data by Equation (3).

Algorithm 1: Real-Time Auction-Algorithm

Input: α_k, γ, δ
Output: x_{ij}
for *request* i *from an online stream* **do**
  $J \leftarrow \emptyset$;
  $L_i \leftarrow$ *bidding landscape generated for* i;
  while $(generate\ L_i^j) and (L_i^j \neq \emptyset)$ **do**
    $J \leftarrow J \cup L_i^j$
  end
  $j' \leftarrow argmax_{j \in J}(rev_{ij} - \sum_{k \in \Omega_{(i,j)}} \alpha_k cost_{ijk} + \gamma \sum_{k \in C_1} ctr_{ijk} + \delta \sum_{k \in C_2} cvr_{ijk})$;
  if $(rev_{ij'} - \sum_{k \in \Omega_{(i,j')}} \alpha_k cost_{ij'k} + \gamma \sum_{k \in C_1} ctr_{ij'k} + \delta \sum_{k \in C_2} cvr_{ij'k}) \geq 0$ **then**
    $x_{ij'} = 1$;
    $x_{ij} = 0, \forall j \neq j'$;
  else
    $x_{ij} = 0, \forall j$;
    no ads displayed;
  end
end

4. Implementation

There are some practical problems that must be considered and overcome in practice. These include metric prediction, performance challenges of online serving system and offline simulation system.

4.1. Metric Prediction

We first introduce ctr_{ijk} which is the estimated click-though rate of ad from campaign k in slate j for request i. Each ad has a specific position p in slate j which means that ctr_{ijk} can be denoted as ctr_{ij_pk}. In consideration of position bias, we estimate ctr_{ij_pk} with *Examination Hypothesis*. The Hypothesis Craswell et al. (2008); Dupret and Piwowarski (2008); Chapelle and Zhang (2009) says that users are more likely to click the first rank item and less likely to look at items in lower ranks, which suggests that each rank has a certain probability of being examined. We have ctr_{ij_pk} with denoting this probability as $P(e|p)$:

$$ctr_{ijk} = ctr_{ij_pk} = P(e|p) \cdot P(c|i,k) \tag{4}$$

where $P(c|i,k)$ is the click-through rate of ad from campaign k assigned to request i. Assuming under the PPC (pay per click) advertising model, similarly, we have the cost of campaign k assigned to request i in slate j:

$$cost_{ijk} = cost_{ij_pk} = ctr_{ij_pk} * clickprice_{ij_pk} \tag{5}$$

The real cost that advertiser should pay when yielding a click is determined by the GSP (generalized second price) mechanism Edelman et al. (2007) in our ad serving system. When the slate j is fixed, we can easily calculate the click price and the expected cost of campaign k assigned to request i.

Assuming the independence of CTRs on the same page, we can express rev_{ij} of request i and slate j in the algorithm as the sum of all the individual expected cost per exposure (*sum method*):

$$rev_{ij} = \sum_{p=1}^{P} rpm_{ij_p k} = \sum_{p=1}^{P} ctr_{ij_p k} \cdot clickprice_{ij_p k} \tag{6}$$

This may not always be a valid assumption, while it is easy to implement and efficient. Zhang et al. (2018) proposed an *interactive method* that try to estimate rev_{ij} more accurately. The method takes into account the interactive influences across ads on the same page. However, due to the huge number of slates generated from bidding landscape, the *interactive method* brings a great challenge of system performance what we will discuss in the next subsection.

4.2. Slate Generation

The intuitive solution of generating slates is to delete some ads in the bidding landscape L_i which will yield at most $(\binom{P}{P_i} + P)$ of slates. The time complexity of this solution is $\mathcal{O}(\binom{P}{P_i} + P)$ which brings a great challenge of the offline training and online serving. Here we propose a simple method to solve the problem based on the *sum method* mentioned in Section 4.1.

In Algorithm 1, we choose the best slate with the highest score of:

$$rev_{ij} - \sum_{k \in \Omega_{(i,j)}} \alpha_k cost_{ijk} + \gamma \sum_{k \in C_1} ctr_{ijk} + \delta \sum_{k \in C_2} cvr_{ijk} \tag{7}$$

which means that the score of every slate must be calculated. We use *sum method* to approximate Eq.(7) as follows:

$$\begin{aligned} &\sum_{p=1}^{P} (rpm_{ij_p k} - \alpha_k cost_{ij_p k} + \gamma I_{C_1}(k) + \delta I_{C_2}(k)) \\ &I_{C_1}(k) = \begin{cases} ctr_{ij_p k}, & \text{if } k \in C_1 \\ 0, & \text{else} \end{cases} \\ &I_{C_2}(k) = \begin{cases} cvr_{ij_p k}, & \text{if } k \in C_2 \\ 0, & \text{else} \end{cases} \end{aligned} \tag{8}$$

Then we can calculate the value of $(rpm_{ij_p k} - \alpha_k cost_{ij_p k} + \gamma I_{C_1}(k) + \delta I_{C_2}(k))$ of each ad in the bidding landscape L_i. We sort the ads in L_i by this score and choose the top P ads to construct the best slate (while maintaining the ordering in L_i). The time complexity is reduced to $\mathcal{O}(P \log(P))$ where P is commonly a small number. It's worth noting that, all these prediction values like $rpm_{ij_p k}$ are calculated based on the order of the original landscape so that there might be a slight inaccuracy by the influence of position bias and GSP mechanism mentioned in subsection 4.1.

4.3. Offline Simulation System

In this work, we need to get the landscape without budget constraints for training and approximate the results by different approaches to measure their performances. We build an offline replay system based on the real ad serving system to obtain training data. Compared with applying to real online system, build a replay system can break out budget constraints and get all the candidate ads.

Our replay system records each request with the traffic information as well as auction information including estimated ctr, cvr and click price of each candidate. The raw data is organized as a table partitioned by time, which amounts to dozens of TBs each day. Based on the replay data from real traffic, we can apply any allocation method on the simulation system to obtain the ads displayed in the page.

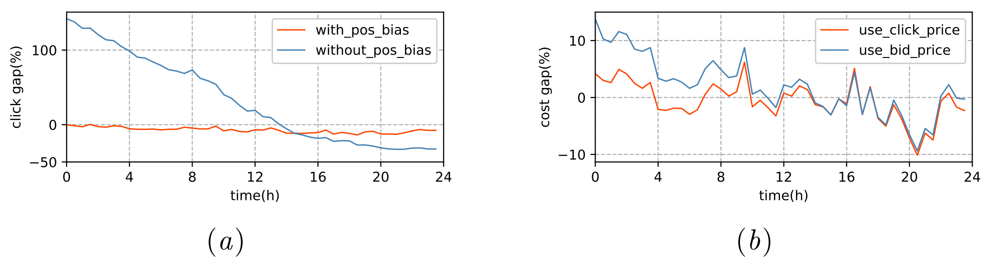


Figure 1: the click and cost gap between simulation system and online system with different ranking and pricing schemas

Furthermore, we try to make the results of online ad serving system and offline simulation system more consistent by improving ranking and pricing schemas. Fig. $1(a)$ demonstrates the click gaps between online and offline system. It shows that the result with considering position bias dramatically outperforms better than that without position bias. Additionally, we calculate real click price by GSP instead of using original bidprice, which makes obvious improvement shown by time in Fig. $1(b)$.

5. Experiments

5.1. Setup

In this section, we conduct some comprehensive experiments to validate the effectiveness of the proposed algorithm. It's worth noting that our approach can be applied to both displaying and search advertising. Here we focus on one of our search advertising scenes and run experiments on the simulation system using real data from Alibaba.com. We choose two days data in May 2019 for training and testing respectively. The dataset covers nearly one hundred thousand campaigns and request data that contains tens of millions of records with auction information such as predicted *ctr*, *cvr*, *click price* and other detail information.

In this paper, we demonstrate two types of experiments and analyze their results:

- **Single-Objective(SO)**: Single-Objective optimization with budget constraints is a specific case of our model. We run the experiments of click optimization which is a common performance goal in E-commerce scenario, and compare the results with previous work Karande et al. (2013).
- **Multi-Objective(MO)**: We try to maximize the platform revenue and improve different performance goals simultaneously. We randomly choose 70% of campaigns to improve their clicks, and 30% of campaigns to increase their conversions. We leverage T_{cy} and T_{vy} as a controller and search the parameter space to get different results of optimal solutions.

We also introduce two different approaches as follows:

- **Greedy Heuristic Policy (GHP)**: Let advertisers participate in auctions as many as possible until they hit their budget and then make them ineligible for the rest of the day. All the relative improvement performance are calculated relative to the **baseline** of the **GHP**.
- **Optimized Throttling (OT)**: Given the objective, we rank all the requests for a campaign by its' performance goal, and choose the request with the top values until the budget is exhausted. They used the performance value of the last request chosen as a threshold to filter the real-time request. The algorithm detail is introduced in Karande et al. (2013).

5.2. Results of Single-Objective

In this subsection, we discuss the experimental results of click and conversion optimization with budget constraints respectively. The results are compared with **OT** Algorithm in Table 1, and the improvements are all calculated relative to the baseline of **GHP**.

The results in Table 1 reveal that our algorithm **SO-clk** dramatically outperforms **GHP** and **OT-clk** in CLK. Compared with **OT-clk**, our approach is also better in REV. As mentioned in Section 2.0, **OT** doesn't work well in a complex competition environment in E-commerce advertising and may cause unnecessary competitions. Besides the overall results, the performance of each campaign is also what we concerned. CLK+ represents the ratio of campaigns whose click number is increased while CLK- is the ratio of campaigns whose click number is declined.

Table 2 shows the results of conversion optimization. **SO-cvn** outperforms **GHP** and **OT-cvn** in both REV and CNV. And more campaigns get better performance in CVN by **SO-cvn** than **OT-cvn** and **GHP**.

Table 1: Relative performance and ratio of campaigns compared with OT-clk

Algorithm	Δ_{CLK}	Δ_{REV}	CLK_+	CLK_-
GHP	-	-	-	-
OT-clk	7.08%	-5.93%	32.29%	46.53%
SO-clk	**28.63**%	-4.95%	**56.61**%	24.71%

Table 2: Relative performance and ratio of campaigns compared with OT-cvn

Algorithm	Δ_{CVN}	Δ_{REV}	CVN_+	CVN_-
GHP	-	-	-	-
OT-cvn	13.71%	-7.66%	26.5%	52.4%
SO-cvn	**39.65%**	-5.85%	**46.33%**	34.98%

Table 3: Relative performance compared with OT-clk-cvn

Algorithm	Δ_{REV}	Δ_{CLK}	$\Delta_{CLK_{C1}}$	Δ_{CVN}	$\Delta_{CVN_{C2}}$
GHP	-	-	-	-	-
OT-clk-cvn	-5.46%	5.80%	10.20%	9.78%	26.6%
MO-T_{cy}15%, T_{vy}30%	**5.61%**	12.55%	**13.91%**	18.00%	**28.25%**

Table 4: The ratio of campaigns

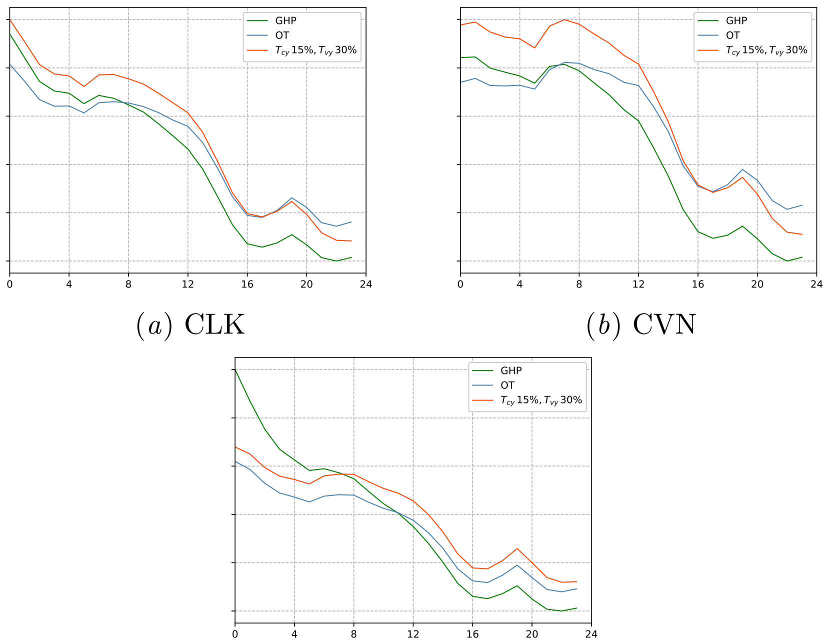
Campaign Ratio	CLK_+	CLK_-	CVN_+	CVN_-
GHP	-	-	-	-
OT-clk-cvn	46.13%	32.68%	43.69%	34.92%
MO-T_{cy}15%, T_{vy}30%	**60.46%**	19.80%	**64.08%**	15.89%

5.3. Results of Multi-Objective

Here we try to maximize the overall revenue and improve advertisers' performance goals (clicks and conversions) simultaneously. We randomly choose 70% of campaigns ($C1$) to improve their clicks, and 30% of campaigns ($C2$) to increase their conversions. Besides, we leverage T_{cy} (the click constraint of $C1$) and T_{vy} (the conversion constraint of $C2$) as a controller and search the parameter space to get different results of optimal solutions. Our algorithm is compared with **OT**, and the results relative to **GHP** are summarized in Table 3 and Table 4.

Δ_{CLK} in Table 3 represents the increase of total clicks of all the campaigns relative to **GHP**, and $\Delta_{CLK_{C1}}$ refers to click increment of the campaigns with click goal (70% of campaigns trying to improve their clicks). Likewise, Δ_{CVN} is the total conversion increment while $\Delta_{CVN_{C2}}$ is the result of the campaigns with conversion goal. Here T_{cy}15% in Table 3 stands for the 15% percentage of improvement on the click of the baseline **GHP** with $C1$ in the train data. And T_{vy}30% represents the 30% percentage of improvement on the conversions of the baseline **GHP** with $C2$. Apparently, our algorithm gets better performance than **OT-clk-cvn** and **GHP** not only in the overall campaigns but also the target campaigns since our algorithm optimizes all requests throughout the day while **OT** only applies to the head queries. Although our model can improve the overall performance, it can not guarantee that the performance of every campaign is improved. The results in Table 4 show that most campaigns with performance goals gets better.

Furthermore, we split the data into 48 time periods (30 minutes between two consecutive time intervals) and study the result of different algorithms on the measures in consecutive time periods in Fig. 2. The curves show that our algorithm sacrifice the revenue in exchange for the click and conversion gain at the beginning of the day. However the overall performances of our method get better than **OT** and **GHP**.



(a) CLK

(b) CVN

(c) REV

Figure 2: CLK, CVN and REV over time with different algorithms

6. Conclusion

In this paper, we focus on optimizing budget constrained spend in E-commerce performance-based advertising. The objective of advertisers is usually to spend out the budget to maximize various performance goals while the ad serving system aims to optimize revenue on behalf of the platform. We propose an approach based on linear programming and build a simulation system to solve the problem. We run large-scale real data experiments in Alibaba.com to demonstrate the benefit of using this algorithm. The results show that it can significantly achieve delivery goals and improve the overall revenue of the platform.

Our future work will mainly focus on incorporating real-time data into the problem setup to improve accuracy. In addition, the offline simulation system needs to be improved.

Acknowledgments

We would like to thank the thank Alibaba.com Ads Engineering team for helpful engineering support.

References

Zoe Abrams, Ofer Mendelevitch, and John Tomlin. Optimal delivery of sponsored search advertisements subject to budget constraints. In *Proceedings of the 8th ACM conference on Electronic commerce*, pages 272–278. ACM, 2007.

Deepak Agarwal, Souvik Ghosh, Kai Wei, and Siyu You. Budget pacing for targeted online advertisements at linkedin. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1613–1619. ACM, 2014.

Anand Bhalgat, Jon Feldman, and Vahab Mirrokni. Online allocation of display ads with smooth delivery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1213–1221. ACM, 2012.

Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10. ACM, 2009.

Peiji Chen, Wenjing Ma, Srinath Mandalapu, Chandrashekhar Nagarjan, Jayavel Shanmugasundaram, Sergei Vassilvitskii, Erik Vee, Manfai Yu, and Jason Zien. Ad serving using a compact allocation plan. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 319–336. ACM, 2012.

Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1307–1315. ACM, 2011.

Alexey Chervonenkis, Anna Sorokina, and Valery A Topinsky. Optimization of ads allocation in sponsored search. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 121–122. ACM, 2013.

Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94. ACM, 2008.

Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 29–38. ACM, 2011.

Georges E Dupret and Benjamin Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 331–338. ACM, 2008.

Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review*, 97(1):242–259, 2007.

Di He, Wei Chen, Liwei Wang, and Tie-Yan Liu. A game-theoretic machine learning approach for revenue maximization in sponsored search. In *IJCAI*, pages 206–212, 2013.

Chinmay Karande, Aranyak Mehta, and Ramakrishnan Srikant. Optimizing budget constrained spend in search advertising. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 697–706. ACM, 2013.

Kuang-Chih Lee, Ali Jalali, and Ali Dasdan. Real time bid optimization with smooth budget delivery in online advertising. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, page 1. ACM, 2013.

Aranyak Mehta et al. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.

Jian Xu, Kuang-chih Lee, Wentong Li, Hang Qi, and Quan Lu. Smart pacing for effective online ad campaign optimization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2217–2226. ACM, 2015.

Weinan Zhang, Shuai Yuan, and Jun Wang. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1077–1086. ACM, 2014.

Weiru Zhang, Chao Wei, Xiaonan Meng, Yi Hu, and Hao Wang. The whole-page optimization via dynamic ad allocation. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1407–1411. International World Wide Web Conferences Steering Committee, 2018.

Yunzhang Zhu, Gang Wang, Junli Yang, Dakan Wang, Jun Yan, and Zheng Chen. Revenue optimization with relevance constraint in sponsored search. In *Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising*, pages 55–60. ACM, 2009.