ROI Constrained Optimal Online Allocation in Sponsored Search

Anonymous

ABSTRACT

Sponsored search plays a major role in the revenue contribution of e-commerce platforms. Advertising systems are designed to maximize platform revenue by displaying relevant ads, also obligated to balance other key performance indicators (KPIs), such as user experience, advertiser utility, and long-term revenue goal. A key component of a sponsored search system is online allocation, which makes real-time decisions to match users' search requests with relevant ad campaigns to maximize platform revenue. Although much progress has been made, most of the research work on allocation problem has focused on guaranteed display ads, and those challenges for allocation problems in auction based sponsored search are not properly addressed. In this paper, we develop a parameter-server architecture based model (ROAM) to solve the large-scale sponsored search ad allocation problem that takes advertiser Return On Investment (ROI) and user experience into consideration, and an online strategy to alleviate the conflict with the auction mechanism during online service. Results of comprehensive offline evaluation on real production data and online A/B testing on real production system demonstrate that through better allocating user queries to appropriate ads, the proposed model can significantly increase the platform's revenue and advertiser's ROI. Moreover, the solution in the paper can be easily generalized to other scenarios where the goal is to maximize user engagements and platform revenue with budget and ROI constraints, for instance, offering incentives to users (coupons at Amazon, discounts at Uber and video bonuses at Tiktok).

CCS CONCEPTS

• **Computing methodologies → Parallel algorithms**; • **Applied computing → Multi-criterion optimization and decision-making**.

KEYWORDS

advertising system, allocation model, ROI, auction mechanism

ACM Reference Format:
Anonymous. 2018. ROI Constrained Optimal Online Allocation in Sponsored Search. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Sponsored search has always been an important part of e-commerce platform revenue generation. When a user issues a search query, search engine returns the user organic search results along with sponsored ads on the same page. In this advertising system, platforms are incentivized to show ads that best match a user's interests with advertiser's bidding keywords, since platforms typically only get paid when a user clicks on an ad. Advertising systems are designed to maximize platform revenue by displaying relevant ads, also obligated to balance other KPIs, such as user experience, advertiser utility, and long-term revenue goal.

A typical sponsored search system is shown in Fig. 1. Advertisers first place an order on the platform by setting target ad-words, target user group attributes, and desired bid and budget settings. In the online service stage, a candidate ads list is determined according to the match with the user's search request (such as search query matching ad-words, etc.), and the subsequent prediction module will output the predicted Click-Through Rate (pCTR) and the predicted post-click ConVersion Rate (pCVR) of each ad. After that, an automatic bid optimization module modifies the bid price to maximize platform revenue and other KPIs. Automatic bidding modification requires the authorization of advertisers, and more than half of advertisers prefer to use fixed bids rather than automated bidding service provided by the platform in most ad systems. Next, the online allocation module, which is the focus of this paper, sets the eligibility of each ad to participate in the auction according to the allocation model trained offline. Finally, the ads participating in the auction are sorted in descending order according to their estimated Cost Per Mille (eCPM = pCTR × Bid) value under the Generalized Second Price mechanism (GSP), and the top ranked k ads are displayed to the user. If an ad in position r is clicked, the advertiser will be charged with the bid price for ad in position $r + 1$.

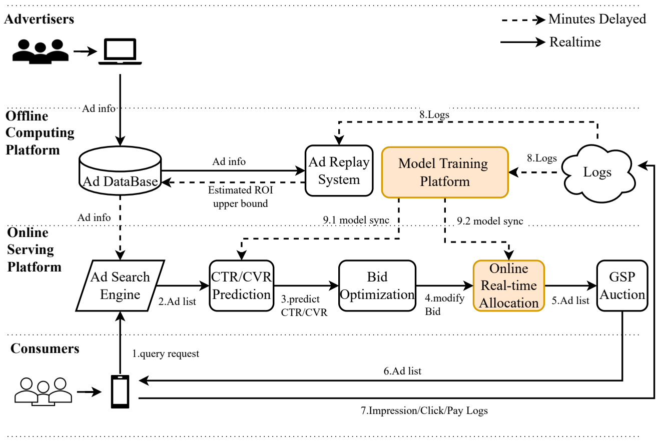

Figure 1: A typical sponsored search system.

One of the key components of the sponsored search advertising system is the online allocation module, which maximizes platform revenue by better matching users' search requests with relevant

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

advertising campaigns in real time, while subject to some additional business constraints, such as campaign budget constraints.

Most of the previous studies on online allocation module are in the area of guaranteed display advertising[3, 4, 6, 8]. Few studies has been done for the challenges of sponsored search [12]:

- **User Experience**. Since search ads are less relevant to users' search intent than organic search results, limiting the number of displayed search ads will benefit the platform from a better user experience standpoint.
- **Long-term Revenue**. In the real advertising system, expectation management is very important. In order to maximize long-term revenue, we need to ensure that the ROI of advertisers will not decrease for a long time. The ROI expectations of advertisers must be met, but we should not optimize the ROI of advertisers to a high level when the number of advertisements is small at the beginning. Starting with high ROI, as the number of advertisements increases, the competition will become fierce, and the ROI of the advertiser will drop, which will lead to the complaint of the advertiser and reduction in advertising budget.
- **Conflict**. There is a conflict between allocation model and Generalized Second-Price (GSP) auction mechanism: allocation model ranks ads with allocation probability while the GSP auction ranks ads with eCPM, which makes allocation result have no effect on advertising system.

In this paper, we formulate the sponsored search ads allocation problem as a constrained optimization problem. In addition to typical advertiser campaign budget constraints, we also consider advertiser ROI constraints, where the lower bound satisfies the advertiser's goals and the upper bound ensures the stability of the advertising ecosystem. Since displaying too many ads will impact users' search experience [9], in addition to platform revenue, we directly put maintaining a certain level of user experience as part of our target functionality. To solve the problem efficiently, a parallel optimization algorithm is developed based on the parameter server architecture, and generates a compact allocation plan for online serving. For the conflict between allocation model and GSP auction mechanism, an online strategy is designed. Comprehensive experiments have been conducted both offline and online on the real production data demonstrating that the proposed model can achieve significant improvements in both advertising platform's revenue and advertiser's ROI.

The main contributions of our work are summarized as follows:

- We propose a new allocation model (ROAM) that simultaneously maximizes platform revenue and minimizes ad impressions, and takes advertiser ROI constraints into consideration.
- We develop a parallel optimization algorithm based on parameter server framework to solve ROAM efficiently, and design an online serving strategy that resolves the conflict between allocation model and GSP auction mechanism.
- In online and offline experiments, our method improves revenue significantly compared to previous methods without sacrificing ROI. After more than three months from the first launch in production environment, our method is still running stably and works efficiently.

2 RELATED WORKS

In online advertising area, allocation algorithms are used mainly for two different purposes: one for optimizing the performance of individual campaign (locally) and one for optimizing the performance of all campaigns on the ad platform (globally).

When optimizing locally, most commonly used allocation algorithm is throttling based method. Throttling-based algorithm usually sets a threshold of ad quality (e.g., click through rate) based on the classic PID algorithm, where campaigns' with quality exceeding the threshold would participate in the auction. Its goal is to throttle the allocation of campaigns whose budget has been consumed too fast [2, 10, 13].

When optimizing globally, allocation algorithm is typically formulated as a constrained optimization problem based on graph matching [11]. Please refer to [11] for a detail and comprehensive survey. [7] find optimal ads allocation in sponsored search by relaxing the original integer programming to a continues optimization problem. Different approaches are proposed to deal with allocation problems with different objectives. For instance, [1] formulates the allocation problem as a Linear Programming (LP) problem and applies column-generation method to solve it. However, it has limited scalability and can only be applied to high frequency queries; [15] proposes a model to minimize the user traffic consumed to satisfy all advertisement contracts. Their method is based on finding the max flow solution for a bipartite graph matching problem. [5, 12] reduce the number of variables of LP problem using dual problem transformation and obtain optimal solutions through offline simulations with historical data to maximize revenue and other KPIs. High Water Mark (HWM) [4] and SHALE [3] are both allocation models proposed for guaranteed display ads. They try to minimize under-delivery penalty, the gap between allocation probability and supply-demand ratio through iterative offline optimization method. [6, 8, 14] extend SHALE to address large scale allocation problem. In addition to guaranteed delivery, they also consider other types of real business needs, such as optimizing click-through rates, penalizing over-allocation, and meeting frequency requirements.

Inspired by previous work above, we design a scalable allocation model for sponsored search. It combines two goals together, one is to maximize advertising revenue and the other is to limit user experience degradation caused by displaying ads. Regarding constraints, it includes not only budget constraint but also ROI constraint to keep the stability of advertiser's ROI.

3 ALLOCATION PROBLEM

3.1 Bipartite Graph of Supply and Demand

The ad allocation problem is usually modeled as a bipartite graph matching problem to maximize revenue with some constraints [3, 14] as illustrated in Fig. 2. Let $G = (I \cup J, E)$ be a bipartite graph, where there are two types of nodes: the supply nodes $i \in I$ that represent user's search request and the demand node $j \in J$ that represent the ad campaign. One supply node is connected with a demand node if the user's search query matches the campaign's target ad words. Each supply node i has a weight s_i indicating its ad capacity and each demand node j has a budget d_j set by the advertisers. The $pCTR_{ij}$ is predicted click-through rate, $pCVR_{ij}$ is predicted post-click conversion rate, $pCPC_{ij}$ is the cost per click

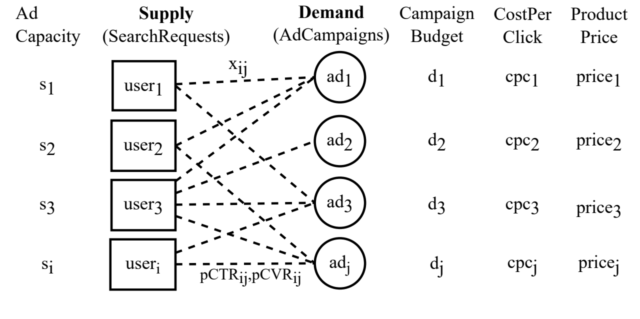

Figure 2: Bipartite graph of supply and demand in sponsored search ad system.

charged to advertisers and $price_j$ is the price of the product/service sold by the advertiser. Since CPC_{ij} is not available prior to GSP auction, we use average historical CPC_j of ad j as $pCPC_{ij}$ instead. In our allocation problem , if one impression from supply node i is allocated to ad j on the demand side, the platform would charge the advertiser c_{ij} and the advertiser would gain Gross Merchandise Volume(GMV) from potential sales g_{ij}, where $c_{ij} = pCTR_{ij} \times pCPC_{ij}$ and $g_{ij} = pCTR_{ij} \times pCVR_{ij} \times price_j$. The origin allocation problem can be defined as :

$$\max_x \sum_{i \in \Gamma(j), j} s_i x_{ij} c_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij} \le d_j, \forall j \qquad \text{(budget constraint)}$$

$$x_{ij} \in \{0, 1\}, \forall i, j \qquad \text{(binary integer constraint)}$$

3.2 A Simple Example of Comparisons between Different Allocation Strategies

An example of allocation problem is shown in Fig. 3. Given ad capacity of the user requests, the pCTR/pCVR for ad_j to be displayed to $user_i$ and the budget/CPC/price of ad_j, the goal of this allocation problem is to find the optimal allocation solution that maximizes revenue as defined in Eq. (1).

Allocation results of different allocation strategies is shown in Fig. 4 to Fig. 6. The bold arrow from ad_j to $user_i$ means $x_{ij} = 1$ which indicates ad_j is picked to display to $user_i$. The revenue of this impression is $eCPM_{ij} = pCTR_{ij} * CPC_j$, and the GMV is $g_{ij} = pCTR_{ij} \times pCVR_{ij} \times price_j$.

In Fig. 4, greedy strategy is used to allocate ad to user requests which greedily choose to display the ad with maximum value of eCPM for each user request. The total revenue of all campaigns is 12, GMV is 18 and ROI is 1.5.

In Fig. 5, short-term revenue maximization allocation strategy is conducted. Comparing to Fig. 4, total revenue increases from 12 to 15.6, while total advertiser ROI decreases from 1.5 to 1.08, especially the ROI of ad_1 decreases from 1.5 to 1 which may lead the advertiser of ad_1 churn on the ad platform.

Comparing to Fig. 4, a balanced allocation strategy shown in Fig. 6 increases revenue of all ads from 12 to 13.1 and increases ROI from 1.5 to 1.76 by taking both revenue and advertiser's ROI into consideration. The improvement of ROI can prompt advertisers to continuously increase their advertising budgets.

Beyond the example, we describe the formulation and implementation of our proposed ROI constrained optimal allocation model which increase revenue and ROI simultaneously for large scale allocation problem in Section 4.

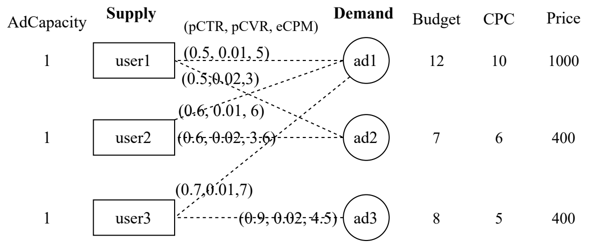

Figure 3: An Example of Allocation Problem .

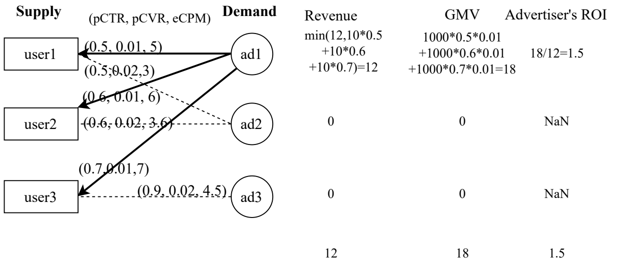

Figure 4: Greedy Allocation Strategy.

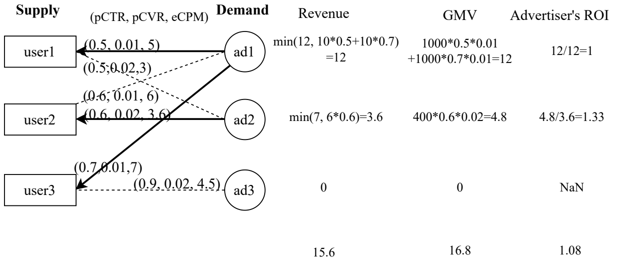

Figure 5: Short-term Revenue Maximization Allocation Strategy.

4 PROPOSED MODEL

4.1 Basic Allocation Problem Formulation

The task of allocation problem is to find the optimal allocation probability x_{ij}, i.e., the fraction of the supply node i allocated to demand node j, that minimizes objectives, and satisfies some constraints. In this paper, we aim to maximize the platform revenue with the minimum ads impressions to minimize the disruption to the organic search results. We also relax the original allocation problem

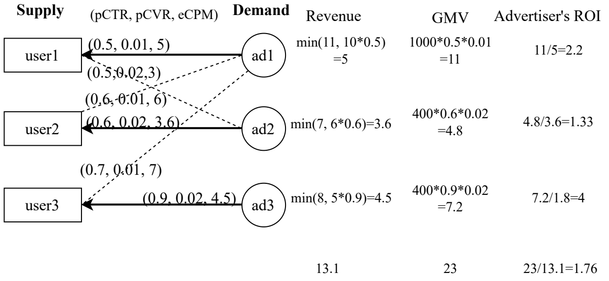

Figure 6: Allocation Strategy that balances the Short-term and Long-term Revenue.

from a binary integer programming to a continues optimization problem[7]. Considering these, the optimal allocation problem in sponsored search can be formally defined as:

$$
\begin{aligned}
\min_{x} \quad & \frac{1}{2} \sum_{i \in \Gamma(j), j} s_i x_{ij}^2 - \lambda \sum_{i \in \Gamma(j), j} s_i x_{ij} c_{ij} && (2) \\
\text{s.t.} \quad & \sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij} \le d_j, \forall j && \text{(budget constraint)} \\
& s_i \sum_{j \in \Gamma(i)} x_{ij} \le s_i, \forall i && \text{(supply constraint)} \\
& u_j \ge \frac{\sum_{i \in \Gamma(j)} s_i x_{ij} g_{ij}}{\sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij}} \ge l_j, \forall j && \text{(ROI constraint)} \\
& x_{ij} \ge 0, \forall i, j && \text{(non-negativity constraint)}
\end{aligned}
$$

where $\Gamma(j)$ is the neighbors of node j in the bipartite graph. The objective is to maximize the platform revenue ($s_i x_{ij} c_{ij}$) while minimizing the impressions allocated ($s_i x_{ij}^2$), i.e minimizing the negative impacts to user experience. User experience is not chosen to be a constraint because there is no clear boundary for user experience. And the square formulation ($s_i x_{ij}^2$) makes it easy to get the solution of x_{ij} using the KKT condition by letting $\frac{\partial L}{\partial x_{ij}} = 0$, then the dual variables can be solved iteratively by coordinate descent or gradient descent algorithm. The hyper-parameter λ balances the revenue and ad impressions.

Constraints. *Budget Constraint:* Each ad campaign has a budget d_j set by the advertiser and the total cost of a campaign should not exceed its budget. *Supply Constraint:* This should be obvious since the total allocation from a supply node i should not exceed its capacity. *ROI Constraint:* The ROI is defined as the ratio of the sales from the ads placed ($\sum_{i \in \Gamma(j)} s_i x_{ij} g_{ij}$) over the cost charged to the campaign ($\sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij}$). As we discussed in Section 1, it is not that the higher ROI is the better for long-term revenue, so ROI is not taken as a part of objective but constraints. Usually, advertisers set up a minimum ROI l_j that we are obliged to guarantee, which implies the least sales revenue that the advertise can achieve with the budget. And We set an upper limit on ROI for each campaign based on the historical data through a replay mechanism. On one hand, the upper limit of ROI is to ensure the stability of the ROI of advertisers, to prevent the ROI of advertisers from being very high when there is no competition, but falling a lot when the competition is fierce. The stability of ROI allows advertisers to make better marketing schedules in advance. On the other hand, the upper limit of ROI is set to reserve some high-quality traffic to improve the performance of ads with extremely low ROI, so as to avoid such advertisers churn on the platform.

4.2 Optimization Algorithm

The allocation problem in Eq. (2) is an optimization problem with convex objective and linear constraints. We can obtain its optimal solution by solving its dual problem through the KKT condition. More specifically, the corresponding Lagrangian function is

$$
\begin{aligned}
L(x, \alpha, \beta, \phi, \eta, \zeta) = & \frac{1}{2} \sum_{i \in \Gamma(j), j} s_i x_{ij}^2 - \lambda \sum_{i \in \Gamma(j), j} s_i x_{ij} c_{ij} \\
& + \sum_j \alpha_j \Big(\sum_{i \in \Gamma(j), j} s_i x_{ij} c_{ij} - d_j \Big) + \sum_i \beta_i \Big(s_i \sum_{j \in \Gamma(i)} x_{ij} - s_i \Big) \\
& + \sum_j \eta_j \Big(l_j \sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij} - \sum_{i \in \Gamma(j)} s_i x_{ij} g_{ij} \Big) \\
& + \sum_j \zeta_j \Big(\sum_{i \in \Gamma(j)} s_i x_{ij} g_{ij} - u_j \sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij} \Big) - \sum_{i \in \Gamma(j), j} \phi_{ij} x_{ij}
\end{aligned} \qquad (3)
$$

From the KKT stationarity condition of $\frac{\partial L}{\partial x_{ij}} = 0$ and the complementary slackness for ϕ_{ij}, i.e., $\phi_{ij} = 0$ unless $x_{ij} = 0$, we have

$$x_{ij} = \max\{0, \lambda c_{ij} - \alpha_j c_{ij} - \beta_i - \eta_j (l_j c_{ij} - g_{ij}) - \zeta_j (g_{ij} - u_j c_{ij})\} \quad (4)$$

which is a function of α_j, η_j and ζ_j, denoted by $x_{ij} = f(\alpha_j, \beta_i, \eta_j, \zeta_j)$. The dual variables α, η and ζ can be solved iteratively by coordinate descend or gradient descent algorithm until the objective function converges. The gradients of α, η and ζ are calculated as:

$$\frac{\partial L}{\partial \alpha_j} = \sum_{i \in \Gamma(j),\ j} s_i x_{ij} c_{ij} - d_j \qquad (5)$$

$$\frac{\partial L}{\partial \eta_j} = l_j \sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij} - \sum_{i \in \Gamma(j)} s_i x_{ij} g_{ij} \qquad (6)$$

$$\frac{\partial L}{\partial \zeta_j} = \sum_{i \in \Gamma(j)} s_i x_{ij} g_{ij} - u_j \sum_{i \in \Gamma(j)} s_i x_{ij} c_{ij} \qquad (7)$$

As public solver (CPLEX, Gurob, etc.) is not free for large scale allocation problems, we propose an efficient parallel algorithm based on Parameter-Server architecture using in-house CPU machines, detailed in Algorithm 1. At first iteration, we calculate β_i with zero as initial values of $\alpha_j, \eta_j, \zeta_j$. After that, in each iteration on worker side, β_i is calculated with equation $\sum f(\alpha_j, \beta_i, \eta_j, \zeta_j) = 1$, and x_{ij} can be obtained with Eq. 4, then on server side $s_i x_{ij} c_{ij}, s_i x_{ij} g_{ij}$ are gathered to update $\alpha_j, \eta_j, \zeta_j$ with Eq. 5 - Eq. 7.

4.3 Online Serving

As shown in Algorithm 2, during the online service process, first, a set of candidate ads that best match user's search request is selected. Then β_i is solved for each request i with equation $\sum_{j \in \Gamma(i)} f(\alpha_j, \beta_i, \eta_j, \zeta_j) = 1$ ($\alpha_j, \eta_j, \zeta_j$ are solved in offline stage). After that, for each ad j in this list, allocation probability x_{ij} can be computed by Eq. (4). Note that ad j^* with highest x_{ij} value among them may not be the winning ads in later auction process, because it may not have the highest eCPM value. However, to maximize platform's revenue, which is proportional to the winning ad's second price, we want ad j^* to

Algorithm 1 Offline Optimal Allocation Algorithm based on Parameter-Server Architecture

```
Input: Demand Side: $d_j$, $u_j$, $l_j (\forall j)$;
       Supply Side: $c_{ij, j \in \Gamma(i)}$, $g_{ij, j \in \Gamma(i)} (\forall ij \in E)$;
Output: the optimal dual values $\alpha_j, \eta_j, \zeta_j, \forall j$
while not converged do
  ▷ Worker:
    for $i \leftarrow 0$ to $|I|$ do
        Calculate $\beta_i$ by solving Equation:
            $\sum_{j \in \Gamma(i)} f(\alpha_j, \beta_i, \eta_j, \zeta_j) = 1$
        if $\beta_i < 0$ or no solution exists then
            update $\beta_i = 0$;
        end if
        for $j \leftarrow 0$ to $|\Gamma(i)|$ do
            Compute $x_{ij}$ with Eq. (4);
        end for
    end for
    Push all $s_i x_{ij} c_{ij}$, $s_i x_{ij} g_{ij}$ to Server;
  ▷ Server:
    Gather all $s_i x_{ij} c_{ij}$, $s_i x_{ij} g_{ij}$ from Worker;
    for $j \leftarrow 0$ to $|J|$ do
        Update $\alpha_j$, $\eta_j$ and $\zeta_j$ with gradients   Eq.(5) - Eq.(7);
    end for
    Synchronize all $\alpha_j$,$\eta_j$,$\zeta_j$ to Worker;
end while
```

Algorithm 2 Online Serving Algorithm

```
1:  for each request $i$ from online stream do
2:      $J \leftarrow \emptyset$;
3:      Calculate $\beta_i$ by solving $\sum_{j \in \Gamma(i)} f(\alpha_j, \beta_i, \eta_j, \zeta_j) = 1$;
4:      if $\beta_i < 0$ or no solution exists then
5:          $\beta_i \leftarrow 0$;
6:      end if
7:      Calculate all $x_{ij}$ with Eq. (4) ;
8:      $j^* \leftarrow argmax_{j \in \Gamma(i)} \{x_{ij}\}$;
9:      if $x_{ij^*} \leq 0$ then
10:         return $\emptyset$ to auction;
11:     end if
12:     $J \leftarrow J \cup j^*$;
13:     $eCPM^* \leftarrow pCTR_{ij^*} * bid_{j^*}$;
14:     for $j \in \Gamma(i)$ do
15:         if $pCTR_{ij} * bid_j < eCPM^*$ then
16:             $J \leftarrow J \cup j$;
17:         end if
18:     end for
19:     return $J$ to participate in the auction;
20: end for
```

be always the winning ads in auction. To solve this conflict between allocation and GSP auction, our allocation algorithm put all ads whose eCPM is lower than ad j^*'s eCPM, to be reserved to participate in the auction, as described in line 8 to line 19 of Algorithm 2.

5 EXPERIMENTS

In this section, we first describe some offline experimental results on production datasets. It includes comparison and analysis of the convergence properties and key performance of different models on these datatsets and a sensitivity analysis of the hyperparameter values in our model ROAM. We also run an online A/B test to test our allocation model's performance at runtime. We use a throttling-based algorithm as a baseline model. The results show that our model outperforms the baseline model in terms of both the platform's RPM and the advertiser's ROI.

5.1 Experimental Settings

Data sets and evaluation metrics. As far as we know , there is no public data set of allocation problem, so we use a real-world data set with 1.2 millions requests(supply node), 622 ad campaigns(demand nodes), and more than 4.8 millions edges in the allocation bipartite graph created from sampling of logs dumped from our production ad system.

For the offline evaluation, we consider the following metrics:

- **Budget Consumption Rate (BCR)** is the ratio of total estimated revenue under allocation to total budget, and how close the offline evaluation result achieved by the methods to the upper bound can be observed by this metric. BCR is calculated as :
$$BCR = \frac{\sum_{i \in \Gamma(j), j} x_{ij} c_{ij}}{\sum_j d_j} \tag{8}$$
- **Revenue Per Mille (RPM)** is a most concerned metric in ad system that represents how much money platform can earn per 1000 impressions. RPM is calculated as :
$$RPM = 1000 * \frac{\sum_{i \in \Gamma(j), j} x_{ij} c_{ij}}{\sum_{i \in \Gamma(j), j} s_i x_{ij}} \tag{9}$$
- **Return on Investment (ROI)** is to directly measure the amount of return on advertiser's displayed ads, relative to the their advertising cost. ROI is calculated as :
$$ROI = \frac{\sum_{i \in \Gamma(j), j} x_{ij} g_{ij}}{\sum_{i \in \Gamma(j), j} x_{ij} c_{ij}} \tag{10}$$

Benchmark methods. For offline allocation algorithm comparison, we compare the performance of our proposed method with four commonly used ones plus a variant version of ROAM, which has excluded the ROI constraint.

- HWM: a geedy method proposed by [4]. It first sorts all contracts in decreasing order of demand-supply ratio, then allocates each contract an equal portion from eligible supplies.
- SHALE: is proposed by [3] , and is modeled to minimize under-delivery penalty and maximize representativeness which is a measure of how close the allocation result is to demand-supply ratio.
- AUAF: is proposed by [6]. It is derived from SHALE with the objective to maximize the contract delivery rate, click through rate and avoid over-allocation.
- ODBC: is proposed by [12], formulates the allocation problem as a single objective linear programming problem to maximize revenue with the constraints of CTR and CVR in

Table 1: Offline Evaluation Results (GMV is short for Gross Merchandise Volume , $GMV = \sum_{i,j} s_i x_{ij} g_{ij}$)

Methods	Revenue	RPM	BCR	GMV	ROI
HWM	21259.32	53.31	13.33%	70015.95	3.2934
SHALE	23291.12	58.40	14.60%	83173.81	3.5692
AUAF	25901.58	64.95	16.24%	94950.61	<u>3.6658</u>
ODBC	**49571.92**	**124.30**	**31.09%**	**173054.56**	3.4910
OAM	<u>40278.77</u>	<u>101.00</u>	<u>25.26 %</u>	135346.46	3.3597
ROAM	39444.78	98.91	24.73%	<u>148571.29</u>	**3.7601**

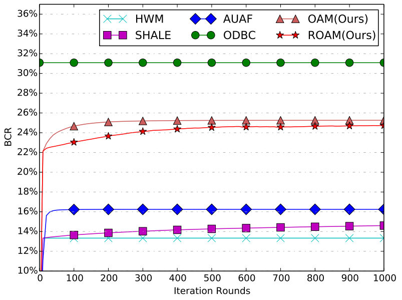

Figure 7: BCR of Offline Allocation.

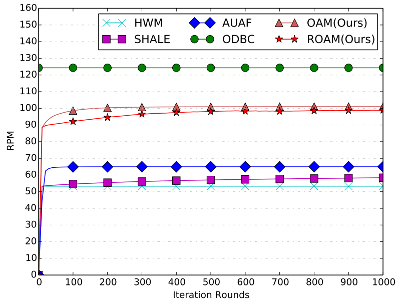

Figure 8: RPM of Offline Allocation.

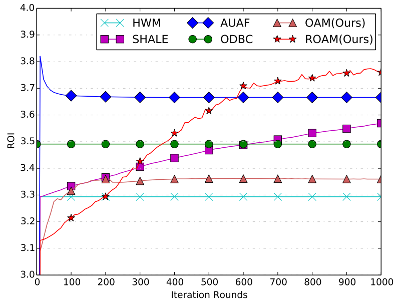

Figure 9: ROI of Offline Allocation.

campaign level. One simplifying assumption is made in this algorithm is that the distribution from which impressions are drawn is stationary, which means given sufficient historical data , optimal priori values of the dual variables can be learned by solving the dual offline.

- OAM (ROAM without ROI constraint) : is to demonstrate the effectiveness of ROI contraints in our proposed model.

For models like HWM, SHALE and AUAF, demand-supply ratio is a key input to the representativeness objective. We have to first convert our ad campaign budgets demand to impressions demand, and calculate the demand-supply ratio, then we can apply these models to our allocation problem.

5.2 Offline Allocation Evaluation

With millions of sample data, final offline evaluation result is displayed in Table. 1; offline convergence is shows in Fig. 7 ,Fig. 8 and Fig. 9; And the influence of hyper parameter λ on our model ROAM is shown in Fig. 10.

Comparison Results. We show the comparison results of different methods on 5 KPIs in Table 1. ODBC's performance is the best on all the indicators except ROI, and it can be regarded as the upper bounds for other methods since it does not consider the ROI constraint. The proposed ROAM achieves the highest ROI while maintaining competitive Revenue, RPM, BCR and GMV. The ROI of AUAF comes to the second place while other metrics show little competitiveness. AUAF is better than SHALE, and SHALE is better than HWM on all metric.

Convergence Analysis. From Fig.7 and Fig. 8, we can see that all the comparison methods converge within 200 iterations on both BCR and RPM metrics. Note that, results for ODBC are shown as a horizontal line since it is solved by open-sourced LP solver without the iteration procedures. HWM is an algorithm that only goes through the data once, hence it is also shown as a horizontal lineand its performance equals that of SHALE after one iteration.

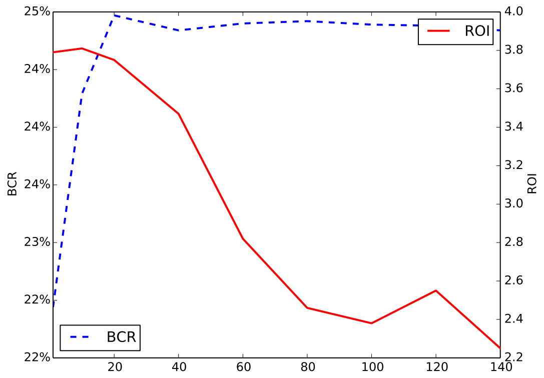

Figure 10: Sensitivity of Hyper-Parameter λ .

Hyper-Parameter Sensitivity Analysis. Larger lambda puts more weight on the revenue and ads will be displayed more times, disrupting user experience. As a result, CVR decreases with more ads shown, so does ROI since $ROI = \frac{price*CVR}{bid}$. Hence, λ is capable to balance ROI and revenue (or BCR). We evaluate the influence of hyper-parameter λ on ROAM in Fig. 10. The experimental optimal hyper-parameter $\lambda = 20$ is obtained by grid search.

Time Consuming With millions of supply nodes and thousand of ad campaigns, our model takes less than 2 hours to train for

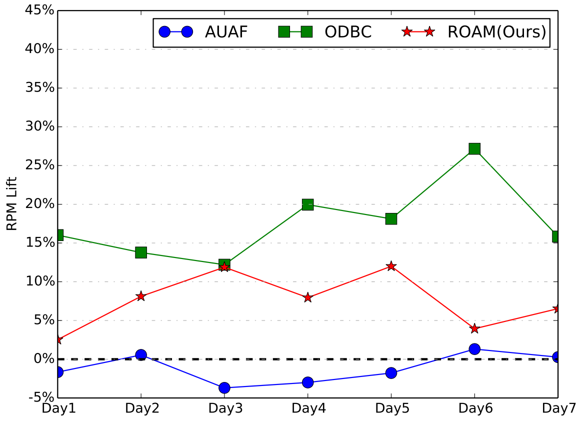

Figure 11: RPM lift of online A/B testing.

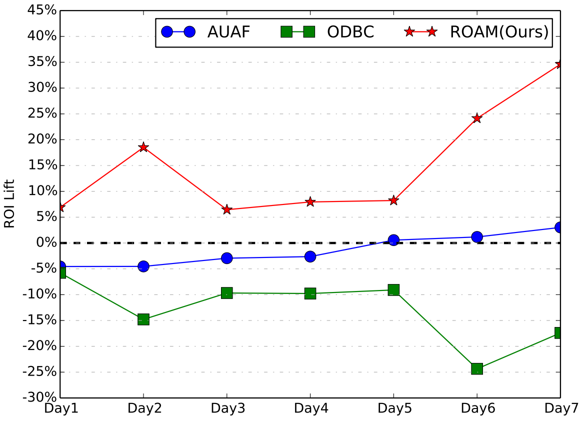

Figure 12: ROI lift of online A/B testing.

1000 epochs with 1 server node (CPU) and 100 worker nodes(CPU). During the online serving stage, the serving time is negligible. The computation time is not a bottleneck.

To summarize, ODBC achieves the best revenue related metrics but with average performance on ROI. Best ROI is achieved by ROAM, followed by AUAF. AUAF is better than SHALE and HWM in terms of budget consumption rate and ROI. We will focus on the online performance of ODBC, ROAM and AUAF in Sec. 5.3.

5.3 Online A/B testing

We conduct the online experiments to compare the proposed ROAM with AUAF, ODBC, and a baseline approach, which is a probabilistic throttling method (as introduced in Section 2) that maximizes the conversion rate from visit to purchase (i.e., pCTR×pCVR) based on the consumption rate of the budget. The online A/B testing is conducted for more than 7 days. Fig.11 and Fig.12 show the comparison results with baseline. Comparing to the throttling-based online baseline method, ROAM is the only method that achieves a positive lift on RPM without sacrificing ROI. Though ODBC has the highest RPM lift, it sacrifices a lot on advertiser's ROI to increase short-term revenue which may lead to advertisers churn. AUAF achieves no lift on both RPM and ROI.

6 CONCLUSION

In this paper, we propose a new reproducible allocation model to optimize short -term and long-term revenue for sponsored search. It consists of two parts. For the offline part, an offline optimal solution is obtained by solving a constrained optimization problem from historical data with a quadratic objective and some linear constraints. It considers both platform revenue and user search experience in its goal, and make a good trade-off between them by setting appropriate hyper-parameter. An iterative algorithm is developed to efficiently solve this optimization problem in large scale. Instead of applying the offline solution directly online, we have designed some online strategy to address the potential conflict between the offline solution and GSP auction mechanism. Both offline and online experimental results show that our new model has made significant improvements on both platform revenue and advertiser ROI. Our future work is to integrate the auction mechanism into the allocation model in a more efficient and elegant way.

REFERENCES

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

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

[3] Vijay Bharadwaj, Peiji Chen, Wenjing Ma, Chandrashekhar Nagarajan, John Tomlin, Sergei Vassilvitskii, Erik Vee, and Jian Yang. 2012. SHALE: An Efficient Algorithm for Allocation of Guaranteed Display Advertising. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China). Association for Computing Machinery, New York, NY, USA, 1195–1203. https://doi.org/10.1145/2339530.2339718

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

[5] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. 2011. 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*. 1307–1315.

[6] Xiao Cheng, Chuanren Liu, Liang Dai, Peng Zhang, Zhen Fang, and Zhonglin Zu. 2021. An Adaptive Unified Allocation Framework for Guaranteed Display Advertising. In *(WSDM)ACM International Conference on Web Search and Data Mining*.

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

[8] Zhen Fang, Yang Li, Chuanren Liu, Wenxiang Zhu, Yu Zheng, and Wenjun Zhou. 2019. Large-Scale Personalized Delivery for Guaranteed Display Advertising with Real-Time Pacing. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 190–199.

[9] Daniel G. Goldstein, R. Preston McAfee, and Siddharth Suri. 2013. The Cost of Annoying Ads. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13)*. Association for Computing Machinery, New York, NY, USA, 459–470. https://doi.org/10.1145/2488388.2488429

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

[11] Aranyak Mehta. 2013. Online Matching and Ad Allocation. *Foundations and Trends in Theoretical Computer Science* 8 (4) (2013), 265–368.

[12] Chao Wei, Weiru Zhang, Shengjie Sun, Fei Li, Xiaonan Meng, Yi Hu, Kuang-chih Lee, and Hao Wang. 2019. Optimal Delivery with Budget Constraint in E-Commerce Advertising. In *2nd Workshop on Online Recommender Systems and User Modeling*. PMLR, 46–58.

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

[14] Hong Zhang, Lan Zhang, Lan Xu, Xiaoyang Ma, Zhengtao Wu, Cong Tang, Wei Xu, and Yiguo Yang. 2020. *A Request-Level Guaranteed Delivery Advertising Planning: Forecasting and Allocation.* Association for Computing Machinery, New York, NY, USA, 2980–2988.

[15] Jia Zhang, Zheng Wang, Qian Li, Jialin Zhang, Yanyan Lan, Qiang Li, and Xiaoming Sun. 2017. Efficient delivery policy to minimize user traffic consumption in guaranteed advertising. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.