Dynamic Reserve Price Design for Lazada Sponsored Search

Mang Li
mang.ll@alibaba-inc.com
Lazada Monetization Tech
Hang Zhou, China

ABSTRACT

In e-commerce platform, users will be less likely to use organic search if sponsored search shows them unexpected advertising items, which will be a hidden cost for the platform. In order to incorporate the hidden cost into auction mechanism which helps create positive growth for the platform, we turn to a reserve price design to decide whether we sell the traffic, as well as build healthy relationships between revenue and user experience. We propose a dynamic reserve price design framework to sell traffic more efficiently with minimal cost of user experience while keeping long-term incentives to the advertisers to reveal their valuations truthfully. A distributed algorithm is also proposed to compute the reserve price with billion-scale data in the production environment. Experiments with offline evaluations and online A/B testing demonstrate that it's a simple and efficient method to be suitably used in industrial production. It has already been fully deployed in the production of Lazada sponsored search.

CCS CONCEPTS

• **Information systems → Computational advertising**.

KEYWORDS

reserve price, incentive-compatibility, large-scale distributed algorithm, sponsored search, advertising

1 INTRODUCTION

Advertising takes an important role in Internet companies like Google and Meta, and it's also the most efficient strategy of monetization in e-commerce platforms such as Amazon, Taobao and Lazada. As we know, building the flywheel to make an e-commerce platform continually grows is to build up a closed-loop among monetization, user growth, and platform traffic & Gross Merchandise Volume (GMV), which shows in figure 1. When money made by ads is invested into user growth, the new or retained users' contribution to GMV can make the platform more valuable to sellers. However, people agree that search ads will hurt user experience and GMV. The first reason for this is the different targets monetization team and organic search team set, the former tries to optimize cost per mille (CPM) while the latter focus on optimizing GMV per mille (GPM). The second reason is that ad items are far less than what are in organic search. But if we keep filling more, the marginal value of ads will decrease, which will finally do harm to the platform. Therefore, building relationship between user experience and revenue when we fill ads is necessary and important. Furthermore, methods that fill ads dynamically are a better way to boost revenue with little loss of user experience. Auction mechanisms, such as reserve price design, can be used to achieve the purpose.

Usually, conversion rate (CVR) or click-through rate (CTR) threshold is used to decide whether we sell the traffic, but it will impose a negative impact on advertisers especially when they are new or put new commodities on shelf because of the uncertainty of estimation. Under the mechanism, no matter how much advertisers raise their bid price, they will not get the traffic and finally quit the campaign. Other methods like [28] try to design a mechanism that encourages advertisers to create an experience for users that maximizes efficiency, but it doesn't work well when ad supply is not sufficient in some traffic because poor quality ads with low bids still have the opportunity for impression. In this paper, we design a framework that generates dynamic reserve price to build the relationship among revenue, user experience, and monetization ratio[1] from the aspect of traffic while keeping the incentive-compatible as well. To compute the reserved price on a large scale of data, we propose distributed algorithm, which can also be used in different problems with the same form.

From the perspective of reserve price design, to maximize the revenue earned in a general second price auction (GSP), the platform can set reserve price and not make any allocations when the bids are low [22]. Lots of work have done to find the optimal reserve price to improve platform revenue from the perspective of advertiser and keep the incentive ability to prevent bid shading issues. A widely used method is based on advertiser's history behavior [14]. In order to estimate the value distribution, the idea is to use advertiser's history of the bids. However, there are two drawbacks of this kind of method, one is that it uses advertisers' original bid to calculate the reserve price, which makes us more careful about the incentive problems. Another one is, for an e-commerce platform, what we sell to the advertisers is the user at different times, which have very different values. It's scarcely possible to estimate the optimal reserve price in practice due to the data sparsity issue. In

---

[1]Monetization ratio defines how much traffic we used to do advertising
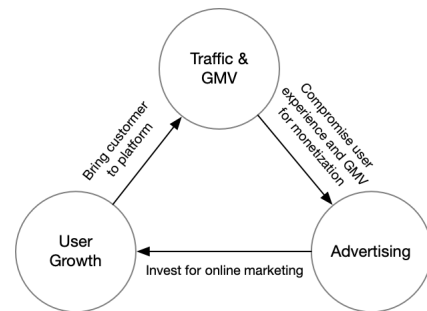

Figure 1: The flywheel to make e-commerce platform continually grow

our proposed method, we rethink the reserve price design from a traffic perspective, and we can prove it's the near-optimal choice to maximize expected cost per mille (ECPM) under different kinds of platform constraints. Moreover, it's also very easy to prove the incentive compatibility property.

Our contributions can be summarized as follows: 1) We design dynamic reserve price to build the relationship between user experience and revenue to make the most of the traffic with incentive compatibility. 2) We proposed distributed algorithm to handle the scalability issues, and it can also be used in problems with the same form. 3) We conducted offline and online experiments to evaluate the effectiveness of the method. In conclusion, it's a very useful method to decide whether we should reserve the traffic to deal with the hidden cost of sponsored search.

2 PRELIMINARIES

A typical description for auction pipeline shows as figure 2, there're M advertisers in sponsored search scenario to compete for N traffic. Due to the intention expressed by the user through a search query, only $K \leq M$ suitable advertisers can participate in one traffic competition. And for each traffic $i, \forall i \in [N]$, using a second price auction, we allocate the traffic to the advertiser who has the highest $ecpm$. we can estimate the $ecpm_i$ to define the potential revenue we can get, where $ecpm_i$ is $bid_i ctr_i$ in cost per click (CPC) model and bid_i is the winner advertiser's bid for traffic i, more specifically, the entire revenue can be denoted as

$$\sum_{i=1}^{N} ecpm_i x_i, x_i \in \{1, 0\} \tag{1}$$

where x_i is the decision variable to decide whether we reserve the traffic, $x_i = 1$ denotes sell otherwise we reserve it. And if there's no other control strategy, the traffic will be filled by ads as much as possible. However, more ads fill means greater potential for compromising the user experience, because the marginal value of ads will decrease.
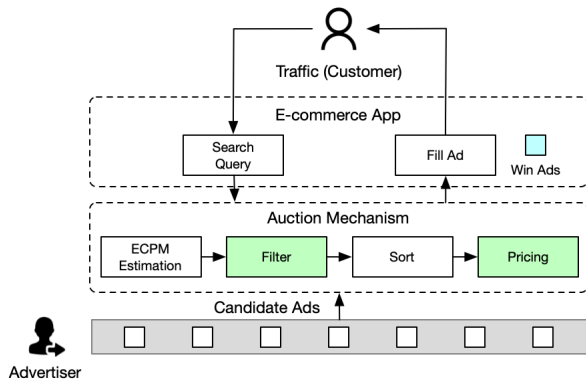

Figure 2: A simple description for e-commerce advertising, and reserve price play a role in stage of filter and pricing in auction mechanism.

2.1 Problem Formulation

As we discuss above, we need to figure out the relation between revenue, user experience, and monetization ratio, so global platform constraints have been added as equation 3-5. We assume that all estimated value is accurate enough to reflect the true probability, which means the CTR can represent the true probability of click-through rate, and the others are similar. We only discuss the reserve price mechanism design without considering the estimation uncertainty. Consider the problem as an optimization problem with platform metrics constraint, such that

$$max \sum_{i=1}^{N} ecpm_i x_i \tag{2}$$

$$s.t. \sum_{i=1}^{N} (ctr_i - tctr)x_i \geq 0 \tag{3}$$

$$\sum_{i=1}^{N} (gpm_i - tgpm)x_i \geq 0 \tag{4}$$

$$\sum_{i=1}^{N} -x_i \geq -tpv \tag{5}$$

$$x_i \in \{0, 1\}, \forall i \in [N] \tag{6}$$

where $tctr$, $tgpm$ means the lower bound of CTR and GPM respectively, which denote user experience constraints, and tpv is the upper bound of ads impression that can also be used to control the user experience. The most important part of the ads platform is the monetization ratio, which is tpv/N and needs to be controlled very cautiously. It's the basic 3 platform constraints, at the same time, we can also add other constraints if it can be written in a linear form.

2.2 Lagrangian Duality

The traffic is very large in sponsored search scenario generally. The scale of the problems we aim to solve should give the solution with distributed cluster rather than a single machine. In this part, we introduce the form of reserve price design with derivation, at first. and then describe the distributed solution.

We rewrite the problem above in a more general form

$$max \sum_{i=1}^{N} c_i x_i \tag{7}$$

$$s.t. \sum_{i=1}^{N} b_{ik} x_i \geq B_k, \forall k \in [L] \tag{8}$$

$$x_i \in \{0, 1\}, \forall i \in [N] \tag{9}$$

where L denotes the constraint size, usually, the constraint is less than 10 in practice. Using the Lagrangian decomposition [16], We define

$$L(x_i; \lambda_k) = \sum_{i=1}^{N} c_i x_i + \sum_{k=1}^{L} \lambda_k (\sum_{i=1}^{N} b_{ik} x_i - B_k) \tag{10}$$

$$= \sum_{i=1}^{N} (c_i + \sum_{k=1}^{L} \lambda_k b_{ik}) x_i - \sum_{k=1}^{L} \lambda_k B_k \tag{11}$$

so we have the Lagrangian relax form and the Lagrange multipliers λ_k. corresponding to the inequality constraints $\sum_{i=1}^{N} b_{ik}x_i \geq B_k, \forall k \in [L]$, λ_k are restricted to be non-negative,

$$\max_{x_i} L(x_i; \lambda_k) \tag{12}$$

$$s.t. \ \lambda_k > 0, \forall k \in [L] \tag{13}$$

$$x_i \in \{0, 1\}, \forall i \in [N] \tag{14}$$

based on the KKT condition [18], the optimal solution for the above problem satisfy

$$\lambda_k(\sum_{i=1}^{N} b_{ik}x_i - B_k) = 0, \forall k \in [L] \tag{15}$$

$$\sum_{i=1}^{N} b_{ik}x_i - B_k \geq 0, \forall k \in [L] \tag{16}$$

$$\lambda_k > 0, \forall k \in [L] \tag{17}$$

So, to maximize $L(x_i; \lambda_k)$ with a fixed λ_k, we only need to select the positive adjust-cost by the decision variable x_i, which can be interpreted as

$$x_i = \begin{cases} 1, & if \ c_i + \sum_{k=1}^{L} \lambda_k b_{ik} > 0 \\ 0, & if \ c_i + \sum_{k=1}^{L} \lambda_k b_{ik} \leq 0 \end{cases} \tag{18}$$

therefore, bringing the problem defined in section 2.1 into equation 18, the dynamic reserve price can be interpreted as follow

$$r_i = [\lambda_1(tctr - ctr_i) + \lambda_2(tgpm - gpm_i) + \lambda_3]/ctr_i, \\ \forall i \in [N] \tag{19}$$

where λ_1, λ_2 and λ_3 is the Lagrangian multiplier [2] for constraint 3-5 respectively

2.3 Distributed Algorithm for Dynamic Reserve Price

2.3.1 Dual Descent Dynamic Reserve Price (D3RP).
First, we consider the dual descent algorithm [21]. According to the decomposability of the Lagrangian dual problem, we can get the solution of the traffic level decision variable parallelly. By using the dual descent algorithm, which is an iterative procedure based on the decomposability of the dual, we can get more optimal Lagrangian multipliers. After we calculate x_i using equation 18, the Lagrangian multipliers can be updated as follow:

$$\lambda_k^{(t)} = max(\lambda_k^{(t-1)} - \alpha(\sum_{i=1}^{N} b_{ik}x_i - B_k), 0) \tag{20}$$

where α is the learning rate for each iteration, and $t \in [0, T]$ denotes the iteration step, where T is the maximum iterations. In algorithm 1, we introduce a distributed version of dual descent to calculate the dynamic reserve price. Particularly, in each iteration. 1) first we use map op in Spark to calculate x_i independently for each traffic, and then we can get the traffic gradient coefficient $b_{ik}x_i$ for each Lagrangian multiplier. 2) we use reduce op to sum all gradient coefficients and collect to the driver to update λ_k using dual descent as equation 20.

---

²λ can be interpreted as the shadow price or marginal utility of the metrics we want to control.

Algorithm 1 Dual Descent Dynamic Reserve Price (D3RP)

1: initialization: $\lambda_k^0 = 0, k \in [L]$
2: **repeat**
3: solve $x_i^{(t)}, i \in [N]$ with multiplier $\lambda_k^{(t-1)}$ by eq 18 in parallel
4: **for** each $k \in [L]$ in parallel **do**
5:   calculate $cons_k^{(t)} = \sum_{i=1}^{N} b_{ik}x_i^{(t-1)}$
6:   update $\lambda_k^{(t)}$ with equation 20
7: **end for**
8: **until** ($\lambda_k^{(t)}$ is converged)

2.3.2 Coordinate Descent Dynamic Reserve Price (CD2RP).
we can see that the algorithm above is simple and efficient enough, however, it requires setting the learning rate α manually, which would be many and diverse. especially, if we try several settings of hyperparameter in computationally intensive scenarios with a large scale of data, it will waste lots of calculation resources. So we propose a method using coordinate descent algorithm [20] to calculate the reserve price, which updated one multiplier λ_k and keeps the other fixed. To avoid the setting of the learning rate, our method prepares an ordered collection of λ_k firstly and then selects the minimal λ_k, which can make the primal problem feasible. We reorganized the subproblem, and write it in the form of a coordinated update as follows

$$max \sum_{i=1}^{N}(c_i + \lambda_{k'}b_{ik'} + \sum_{k=1, k \neq k'}^{L} \lambda_k b_{ik})x_i, \tag{21}$$

$$x_i \in \{0, 1\}, \forall i \in [N], \forall k \in [L] \tag{22}$$

let $f_i(\lambda_{k'}) = c_i + \sum_{k=1, k \neq k'}^{L} \lambda_k b_{ik} + \lambda_{k'}b_{ik'}, \forall i \in [N]$, according to equation 18, the decision variable is only related to the sign of the coefficient function $f_i(\lambda_{k'})$, the interaction with the horizontal axis can make the decision variable change, so the ordered set of λ_k can be defined as

$$\lambda_k \in \{\lambda_{k'}|f_i(\lambda_{k'}) = 0, \forall i \in [N]\}. \tag{23}$$

and we can see that N is a number of billion-scale, the candidate set of λ_k is the same order of magnitude as N. So λ_k search algorithm with time complexity $O(n)$ is unacceptable. We can design dual binary search (DBS) to find the temporary optimal λ_k if the primal problem is feasible.

More specifically, the procedure of the algorithm 2 shows below: 1) we calculate sorted candidates of Λ by equation 23 in parallel, 2) use binary search to find the minimum threshold to guarantee the constraint not less than B_k and then update λ_k.

2.4 Incentive Compatibility Discussion

Incentive compatibility is the most important feature for reserve price design, as the bid of an advertiser may determine the price he pays in future auctions, if we don't design the reserve price with incentive compatibility, it may lead the advertisers to shade their bids and finally result in a loss in revenue for the platform. In this part, we prove the incentive compatibility of our design of reserve price. Consider one advertiser j in an auction of traffic $i, \forall i \in [N]$ and assume that the other advertiser will be truthful.

Algorithm 2 Coordinate Descent Dynamic Reserve Price (CD2RP)

1: initialization: $\lambda_k^0 = 0, k \in [L]$
2: **repeat**
3:    **for** each $k \in [L]$ **do**
4:       compute sorted candidate set $\Lambda_k^{(t)}$ by eq 23 in parallel
5:       search optimal $\lambda_k^{(t)}$ and $x_i^{(t)}, i \in [N]$ with candidate set
      $\Lambda_k^{(t)}$ and algorithm 3
6:    **end for**
7: **until** ($\lambda_k^{(t)}$ is converged)

Algorithm 3 Dual Binary Search (DBS)

1: input: $\Lambda_k^{(t)} = \{\lambda_{k_0}^{(t)}, \lambda_{k_1}^{(t)}, ..., \lambda_{k_{K-1}}^{(t)}\}, B_k$, where $K = |\Lambda_k^{(t)}|$ and
   $k \in [L]$
2: initialization: $l = 0, u = K - 1$
3: **while** $l < u$ **do**
4:    $p = \lfloor (\ell + u)/2 \rfloor$
5:    solve $x_{i_p}^{(t)}, i \in [N]$ with multiplier $\lambda_{k_p}^{(t)}$ by eq 18 in parallel
6:    calculate $cons_k^{(t)} = \sum_{i=1}^N b_{ik} x_{i_p}^{(t)}$ in parallel
7:    **if** $cons_k^{(t)} < B_k$ **then**
8:       $l = p + 1$
9:    **else**
10:      $u = p$
11:    **end if**
12: **end while**
13: return $\lambda_{k_u}^{(t)}, x_{k_u}^{(t)}$

We can see that for each advertiser j, the dynamic reserve price and her prices to win the traffic do not depend on j's own bids at all. Hence, the bid of the advertiser j does not affect his utility in future rounds and myopically maximizing utility in each round is optimal for maximizing the long-term utility. Since truthful bidding is myopically a dominant strategy in each round, being truthful is the best choice, and all advertiser following the always truthful strategy constitutes an equilibrium status. Without loss of generality, let's consider the simple case for single ad slot. Advertiser j uses bid_{ij} to compete traffic $i, \forall i \in [N]$ with his valuation v_{ij}, and the dynamic reserve price for the advertiser j is drp_{ij}, which do not depend on bid_{ij}, and the pay per click (PPC) is $ppc_{ij} = max_{k \neq j} bid_{ik}$ if there's no reserve price. Otherwise, we charge advertiser $max(max_{k \neq j} bid_{ik}, drp_{ij})$, so the traffic i's utility for advertiser j is $u_{ij} = bid_{ij} - max(max_{k \neq j} bid_{ik}, drp_{ij})$. Let's consider the following situations in table 1, we can see that telling the truth i.e. $bid_{ij} = v_{ij}$ is a weakly dominant strategy, which can lead to a truthfully bidding, so this kind of dynamic reserve price design is incentive compatibility.

2.5 Convergence and Optimality Discussion

In general, when the algorithm converges to a result, that jointly satisfies the optimality condition of equation 15-17, and is the optimal solution to the primal problem. However, the Lagrangean techniques can not guarantee to get an optimal solution to the

Table 1: Advertiser incentive discussion with different kinds of competition context

bidding strategy	competition context	utility	optimal utility
$bid_{ij} > v_{ij}$	$bid_{ij} > drp_{ij} > v_{ij} > ppc_{ij}$	< 0	$= 0$
	$bid_{ij} > drp_{ij} > ppc_{ij} > v_{ij}$	< 0	$= 0$
	$bid_{ij} > v_{ij} > drp_{ij} > ppc_{ij}$	> 0	> 0
	$bid_{ij} > v_{ij} > ppc_{ij} > drp_{ij}$	> 0	> 0
	$bid_{ij} > ppc_{ij} > v_{ij} > drp_{ij}$	< 0	$= 0$
	$bid_{ij} > ppc_{ij} > drp_{ij} > v_{ij}$	< 0	$= 0$
$bid_{ij} < v_{ij}$	$v_{ij} > drp_{ij} > bid_{ij} > ppc_{ij}$	$= 0$	> 0
	$v_{ij} > drp_{ij} > ppc_{ij} > bid_{ij}$	$= 0$	> 0
	$v_{ij} > bid_{ij} > drp_{ij} > ppc_{ij}$	> 0	> 0
	$v_{ij} > bid_{ij} > ppc_{ij} > drp_{ij}$	> 0	> 0
	$v_{ij} > ppc_{ij} > bid_{ij} > drp_{ij}$	$= 0$	> 0
	$v_{ij} > ppc_{ij} > drp_{ij} > bid_{ij}$	$= 0$	> 0

primal integer programming (IP) problem, because a duality gap occurs due to the LP relaxation [27]. However, the solution computed from 12-14 is optimal for any IP problems derived from 15-17, when B_k is replaced by $B_k - \delta_k$ [16], where δ_k is non-negative variables, which satisfy for any $\lambda_k > 0$ such that δ_k is 0. But we can still analyze the gap between optimal and our algorithm empirically, the result in table 2 shows that it can achieve a nearly optimal solution. In particular, the optimality gap decreases as the number of traffic increases. Generally, the data scale is quite large in the e-commerce scenario, which means we can get the near-optimal solution in general.

3 EXPERIMENTAL EVALUATIONS

we do the test on both artificial data and Lazada sponsored search scenario data. For artificial setting, we generate c_i and the coefficient of global constraint b_{ik} using two uniformly distribution in $[0, 1]$ respectively, at the same time, without losing generality, we set the constraint B_k between 0 and N, and make it has a feasible solution. For a real scenario on a large dataset, c_i is replaced by $ecpm_i = ctr_i bid_i$, and we test three constraints as described in section 2.1. The optimality is evaluated by duality gap [29], which can quantify the discrepancy between the objective value computed by a current feasible but suboptimal solution of the primal problem and the value calculated by the dual problem.

3.1 Optimality Testing

First, we evaluate the optimality ratio, which is defined as the ratio of objective value solving by SCIP [3] [15] to that using our distributed algorithm. Because it's difficult to find any solver that can solve billion-scale IP problems, therefore, we only test the optimality under medium-scale problems using artificial data generated as described above. The experiment results show in figure 3, which demonstrate that the optimality ratio is above 0.995 with $N = 2000$, and above 0.998 with $N = 10000$ under all scenarios of local constraints for all experiment cases. More details, we compare the

---

[3]SCIP is currently one of the fastest non-commercial solvers for mixed integer programming (MIP) and mixed integer nonlinear programming (MINLP)

two algorithms described above across the different number of constraints, where $1 \leq K \leq 20$ and all coefficients generated by the uniform distribution. We draw the average optimality ratio curve with 20 times test as we vary K, the result shows in Figure 1. The optimality gap increases while K increases, at the same time, decreases while N increases. In practice in industry, we usually have small K but large N, so it proves empirically we can get a near-optimal solution.
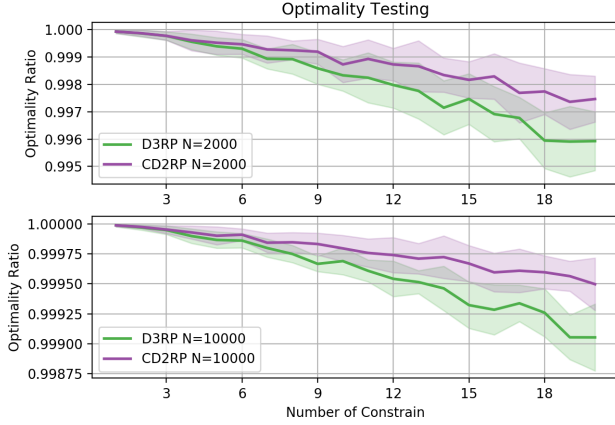

Figure 3: Optimality ratio between solutions computed by SCIP and algorithm by LP relaxation

3.2 Duality Gap on Production Data

we test the algorithm with real search scenario data to measure the optimality of the solution. As it's difficult to find the solution to billion-scale problems, therefore, one of the choices is to calculate the duality gap D combined with primal objective value O, and if the relative duality gap defined as D/O is small enough, it means that we find the near-optimal solution. So, we use it to describe the level of optimization. As described above we test the two algorithms by real data with three constraints, table 2 shows the result of the relative duality gap after 15 iterations, which indicates we have already achieved a near-optimal solution. We can see the relative duality gap curve from figure 4, which shows that D3RP is sensitive to the learning rate while having a slower converge speed than CD2RP.

Table 2: Comparison of relative duality gap after 15 iterations

Iteration	D3RP ($\alpha = 1.5e^{-9}$)	D3RP ($\alpha = 2e^{-9}$)	CD2RP
15	$6.3869e^{-4}$	$5.4895e^{-5}$	$1.5866e^{-9}$

3.3 Solving Speed

The distributed implementation of our algorithm is using Spark, and in order to study the scalability, we try different scales of traffic sampled in real data with N range from million-scale to billion-scale

with fix $K = 3$ (CTR, GPM and PVR), and we set 512 number of executors, where each one with 1 core and 2G memory. Experiment has been tested 3 times in the same setting to get more reliable results. Figure 5 shows the solving time with different scales of traffic. The optimization for 20 million-scale decision variables within $K \in \{1, 3, 5, 7\}$ global constraints can be able to converge to a near-optimal solution within an hour, which shows in figure 6.

3.4 Production Deployment

We present the online performance of the proposed dynamic reserve price design in Lazada sponsored search scenario, in which we consider not only efficiency, such as CTR and GPM, but also the relevance score[4]. We consider four metrics, i.e., Revenue, CTR, GPM,

---

[4]Relevance score is a human evaluation metric defining whether the ads are relevant to the query.
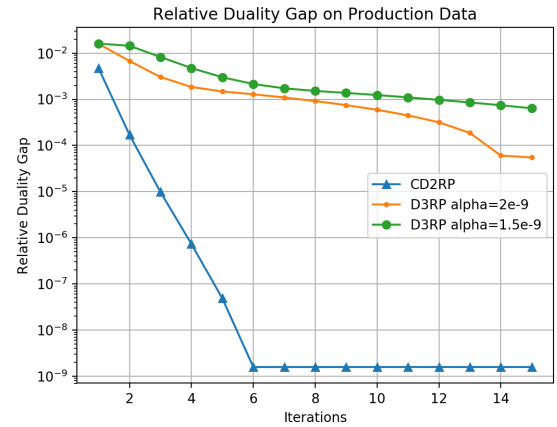

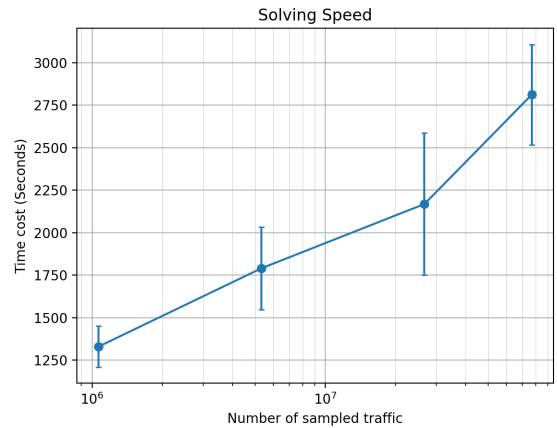**Figure 4: Duality gap curve over iterations for D3RP and CD2RP**

Figure 5: Time cost over different scales of traffic on 3 constraints and 512 executors using CD3RP

top query[5] relevance and mid query[6] relevance, and conduct online A/B testing with these metrics. We define the relevance score to be better than 85% in top query and 75% in mid query, at the same time, CTR and GPM gap of ad slot compared with pure organic bucket should be less than 20% and 30% respectively. Table 3 shows the relative improvements with 10% of production traffic during a week, and the benchmark is the traffic not using any traffic control strategy. From the experiments, dynamic reserve price can make it possible to get high performance of CTR, GPM and relevance with a low cost of revenue. At the same time, as the online platform contains millions of user requests every day, the results can prove stable. The distributed algorithm we developed in this paper has already been fully deployed in production.

Table 3: Online experiment for Lazada sponsored search in Thailand

CTR	GPM	Top Query Relevance	Mid Query Relevance	Revenue
+24.71%	+30.6%	+59.71%	+30.14%	-1.77%

4 RELATED WORK

In this section, we briefly discuss the work related to reserve price design for the application of online advertising. Most of the methods design the reserve price from the advertiser perspective using their history purchase, which differs from our work designing reserve price from the traffic perspective.

the recent study on reserve price strategy is based on Myerson optimal auction [1] and tries to estimate the advertisers' unknown valuations to maximize platform revenue. In practice, Ostrovsky and Schwarz conducted an experiment on reserve prices for online auctions [2], which showed that it can significantly increase

---

[5]Top query is the query with top 50% search request.
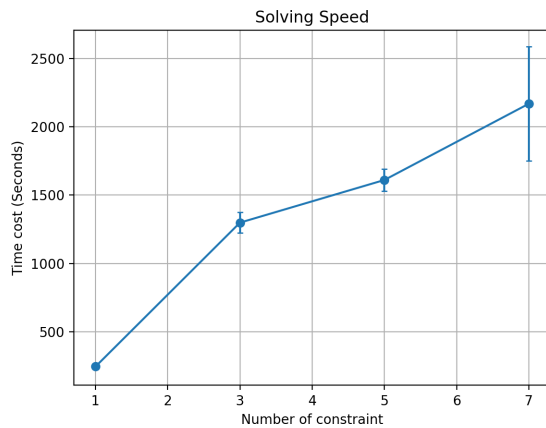[6]Mid query is the query with mid 10% search request.

Figure 6: Time cost over different number of constraint on ten million scale decision variable using CD3RP

the revenue of sponsored search. And a common insight, which is widely used in practice, is to estimate advertisers' valuation distributions using history bids. However, These methods can not bring long-term incentives to the advertisers. Because it may lead them to shade their bids and ultimately hurt platform revenue [6–8].

The rapid development of e-commerce technology in recent years [26], it renewed interest in reserve price design based on advertiser behavior, which is setting a personal reserve price for each advertiser based on their previous bid price to boost revenue [10–14]. Unfortunately, some designs of personalized reserve price still have incentive-compatibility issues, which have been discussed in [14].

Yash Kanoria al. [14] show how behavior base reserve price can be used to optimize reserve prices in an incentive-compatible way by appropriately learning from the previous bids. they propose an incentive-compatible mechanism that sets a personal reserve price for each advertiser based on the previous bids of other advertisers. However, all of these methods above potentially assume that all traffic value is similar to the advertiser, which may not be suitable for an e-commerce scenario.

5 CONCLUSION

We introduce the dynamic reserve price design framework for Lazada sponsored search, it can be used to define the relationship between revenue and user experience with incentive compatibility. We proposed distributed algorithms to solve the billion-scale problem with limited constraints, which can also be used to solve other similar problems in computational advertising. The proposed algorithm can be implemented not only using Spark but also using other distributed frameworks [17] such as MPI and Flink. The implementation in the production environment is simple to guarantee the stability of the whole advertising system.

REFERENCES

[1] Myerson, R.B. Optimal auction design. *Mathematics of operations research* **1981**, *6*, 58–73.
[2] Ostrovsky, M.; Schwarz, M. Reserve prices in internet advertising auctions: A field experiment. In Proceedings of the Proceedings of the 12th ACM conference on Electronic commerce, 2011, pp. 59–60.
[3] Stokey, N.L. Intertemporal price discrimination. *The Quarterly Journal of Economics* **1979**, pp. 355–371.
[4] Salant, S.W. When is inducing self-selection suboptimal for a monopolist? *The Quarterly Journal of Economics* **1989**, *104*, 391–397.
[5] Hart, O.D.; Tirole, J. Contract renegotiation and Coasian dynamics. *The Review of Economic Studies* **1988**, *55*, 509–540.
[6] Villas-Boas, J.M. Price cycles in markets with customer recognition. *RAND Journal of Economics* **2004**, pp. 486–501.
[7] Taylor, C.R. Consumer privacy and the market for customer information. *RAND Journal of Economics* **2004**, pp. 631–650.
[8] Fudenberg, D.; Villas-Boas, J.M. Behavior-based price discrimination and customer recognition. *Handbook on economics and information systems* **2006**, *1*, 377–436.
[9] Bikhchandani, S.; McCardle, K. Behavior-based price discrimination by a patient seller. *The BE Journal of Theoretical Economics* **2012**, *12*.
[10] Amin, K.; Rostamizadeh, A.; Syed, U. Learning prices for repeated auctions with strategic buyers. *Advances in Neural Information Processing Systems* **2013**, *26*.
[11] Acquisti, A.; Varian, H.R. Conditioning prices on purchase history. *Marketing Science* **2005**, *24*, 367–381.
[12] Mohri, M.; Munoz, A. Optimal regret minimization in posted-price auctions with strategic buyers. *Advances in Neural Information Processing Systems* **2014**, *27*.
[13] Chen, X.; Wang, Z. Bayesian Dynamic Learning and Pricing with Strategic Customers. 2016.
[14] Kanoria, Y.; Nazerzadeh, H. Dynamic reserve prices for repeated auctions: Learning from bids. *arXiv preprint arXiv:2002.07331* **2020**.

[15] Achterberg, T. SCIP: solving constraint integer programs. *Mathematical Programming Computation* **2009**, *1*, 1–41.

[16] Shapiro, J.F. A survey of Lagrangean techniques for discrete optimization. In *Annals of Discrete Mathematics*; Elsevier, 1979; Vol. 5, pp. 113–138.

[17] Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters **2004**.

[18] Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex optimization*; Cambridge university press, 2004.

[19] Agarwal, D.; Chen, B.C.; Elango, P.; Wang, X. Personalized click shaping through lagrangian duality for online recommendation. In Proceedings of the Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, 2012, pp. 485–494.

[20] Wright, S.J. Coordinate descent algorithms. *Mathematical Programming* **2015**, *151*, 3–34.

[21] Macklin, M.; Erleben, K.; Müller, M.; Chentanez, N.; Jeschke, S.; Kim, T.Y. Primal/dual descent methods for dynamics. In Proceedings of the Computer Graphics Forum. Wiley Online Library, 2020, Vol. 39, pp. 89–100.

[22] Krishna, V. *Auction theory*; Academic press, 2009.

[23] Lahaie, S.; Pennock, D.M. Revenue analysis of a family of ranking rules for keyword auctions. In Proceedings of the Proceedings of the 8th ACM Conference on Electronic Commerce, 2007, pp. 50–56.

[24] Evans, D.S. The online advertising industry: Economics, evolution, and privacy. *Journal of economic perspectives* **2009**, *23*, 37–60.

[25] Goldfarb, A.; Tucker, C. Online display advertising: Targeting and obtrusiveness. *Marketing Science* **2011**, *30*, 389–404.

[26] Turban, E.; King, D.; Lee, J.K.; Liang, T.P.; Turban, D.C. Marketing and advertising in e-commerce. In *Electronic commerce*; Springer, 2015; pp. 403–456.

[27] Dantzig, G.B. Discrete-variable extremum problems. *Operations research* **1957**, *5*, 266–288.

[28] Abrams, Z.; Schwarz, M. Ad auction design and user experience. In Proceedings of the International Workshop on Web and Internet Economics. Springer, 2007, pp. 529–534.

[29] Champion, T. Duality gap in convex programming. *Mathematical programming* **2004**, *99*, 487–498.

[30] Aggarwal, G.; Muthukrishnan, S.; Pál, D.; Pál, M. General auction mechanism for search advertising. In Proceedings of the Proceedings of the 18th international conference on World wide web, 2009, pp. 241–250.