

Real Time Bidding in Online Digital Advertisement

Shalinda Adikari and Kaushik Dutta

National University of Singapore, Singapore, Singapore
shalinda.adikari@gmail.com, duttak@nus.edu.sg

Abstract. Real time bidding (RTB) is becoming the key to target marketing where it could optimize advertiser expectations drastically. Not like the conventional digital advertising, in the process of RTB, the impressions of a mobile application or a website are mapped to a particular advertiser through a bidding process which triggers and held for a few milliseconds after an application is launched. To carry out the bidding process a special platform called demand side platform (DSP) provides support to advertisers to bid for available impressions on their behalf. This process has turned into a complex mission as there are many applications/websites that have come into the market. Mapping them to advertisers' target audience, and bidding appropriately for them is not a simple human mediated process. The complexity and the dynamic nature in the RTB process make it difficult to apply forecasting strategies effectively and efficiently. In this paper we propose an autonomous and a dynamic strategy for bidding decisions such as bidding price. We applied our proposed approach on a real RTB bidding data and demonstrated that our approach can achieve higher conversion rate with the target spend for a DSP.

Keywords: Real time bidding · Demand side platform · Bid price · Bid request · Impressions · Advertiser expectations · Target audience · Forecasting · Dynamic programming

1 Introduction

Real-time bidding (RTB) is considered as the new era of digital advertising. RTB is autonomous and algorithm driven, which completes a full transaction in milliseconds on pre-set parameters. RTB is a programmatic instantaneous auction, which allows impression buyers to launch their advertising campaigns via multiple ad-networks. It enhances online advertising while providing higher opportunities for advertisers to publish their ads and increasing the publishers' gain through competitive advantage. According to the recently published Online Advertisers Survey Report [5], among 650 advertisers 62% of them see improved performance as the main advantage of RTB; in addition, the trading desk spent on RTB, globally stands at 40%. Mainly, RTB helps to reduce the wastage of manual intervention; also it facilitates better target marketing strategies. Therefore, compared to static auction of digital marketing, RTB can add dynamism to the bidding process where advertisers are bidding for a specific single impression to optimize their expectations.

RTB consists of many different challenges compared to the currently offered buying and selling policies in advertising. Mainly, as the name itself implies, RTB is a real time process that we need to decide our offer within less than 120 milliseconds [11], thus it makes obscure to practice highly complex and time consuming techniques for the decision making. Therefore, this study recommends a novel approach to decide the bid price of the RTB in ad platforms. The proposed model follows a dynamic programming (DP) algorithm to adjust the bid price based on the advertiser's target goals. We have tested our approach on the real mobile RTB campaign data and demonstrated its effectiveness.

1.1 Background

As shown in Fig. 1, RTB ecosystem has two sides, that is, advertiser side and publisher side. Each side has its own components and techniques in the bidding process [11]. The RTB allows purchasing individual impressions through a bid engine that unfolds within milliseconds when an application is launched by a consumer. Each step of the bidding process is demonstrated in Fig. 1 [11, 15].

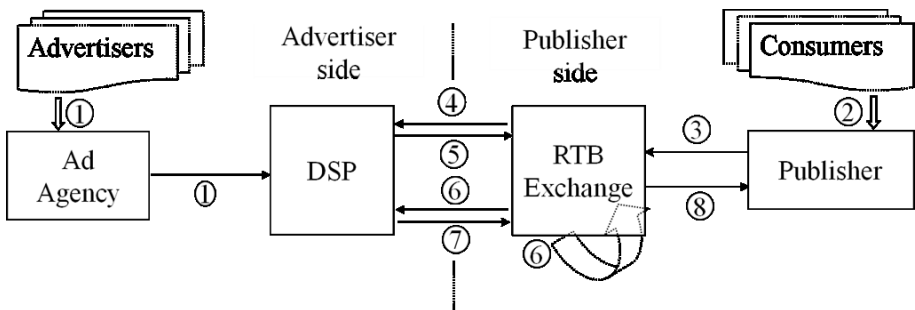


Fig. 1. High-level communication diagram between parties in the Open RTB Ecosystem

The RTB process can be explained further via the following steps,

- Step 1: Advertiser request a DSP to run an ad campaign for a particular product based on predefined campaign budget, target audience and campaign duration.
- Step 2: When the user interaction originates, the mobile app or the web browser sends user preferences, context, location and the mobile device/browser information to the publisher to fill up the impressions in the mobile app or the website.
- Step 3: Then the publisher will check whether there is a contracted advertiser available for the mobile app or the website based on the previous agreement and if that is still valid. Then the ad request is sent to that particular advertiser. If the quota is not available or the contracted advertiser is not interested in the new impression, then the ad request is sent to the RTB exchange.

- Step 4: RTB exchange or sell side platform will create a bid request for the incoming ad request and send it to all subscribed DSPs (See Table 1 for content of a bid request).
- Step 5: DSP will decide the bid price based on an anticipated ad campaign of a particular ad agency. All DSPs send their bid responses with the bid price for the relevant RTB exchange (See Table 1 for the content of a bid response).
- Step 6: After a predefined fixed time period has elapsed, RTB exchange apprehends the auction and decides the bidder who has made the highest bid price as the winner through second bid price auction [6]. Then RTB exchange will send the win note to the winning DSP with the winning bid price (See Table 1 for content of a win note).
- Step 7: DSP requests the ad from the ad agency and sends the ad response to the RTB exchange.
- Step 8: The RTB exchange will forward the advertisement to the germane publisher and then users can see the advertisement on their mobile application or web page.

Table 1.0 provides essential details which are included in the bid request, bid response, win note and conversion note [11]. The bid request “id” is a unique identifier for each bid request which is created by RTB exchange and used throughout the RTB process. Additionally, DSPs create another unique identifier (adid, see Table 1.0) to track ad conversions.

Table 1. Details of the bid request, bid response, win note and conversion note in the RTB process

Field	Description
Bid request (Step 4)	
id	Unique ID of the bid request, provided by the RTB exchange.
timestamp	Time of the bid request initiated
imp	Describes the ad position or impression being auctioned.
site/app	Whether the ad supported content is part of a website or mobile application. Also, it includes information such as identifier, name, Domain, publisher, content and keywords which describe the site/app
Device	Information pertaining to the device including its hardware, platform, location, and carrier.
Geo	Describes the current geographic location of the device (e.g., based on IP address or GPS), or it may describe the home geo of the user (e.g., based on registration data).
User	Describes the user details such as year of birth, gender and user interests
tmax	The maximum amount of time in milliseconds to submit a bid (e.g., generally the bidder has 120ms to submit a bid before the auction is complete)
Bid response (Step 5)	
Id	Relevant bid ID which the response is mapped
Price	Bid value which is decided by the DSP

Table 1. (Continued)

currency	Type of the currency, which the bid is made
nurl	Win notice the URL
adid	An identifier that references the ad to be served if the bid wins and it is stated by the DSP.
Win note (Step 6)	
id	Same identifier as adid in the bid response
winPrice	Winning bid value which is decided by the RTB exchange
currency	Type of the win price currency
Conversion note	
impression_id	Same identifier as adid in the bid response
timestamp	Time which is the conversion happened (the time user views the advertisement)

As per the work flow of publishers and ad agency communication, DSP is bidding on behalf of the ad agency, to buy impressions from applications which have similar target audience as an advertiser is expected. An advertising agency is seen as any third party or in-house team which works on behalf of advertisers to broadcast their advertising performance. An ad exchange is responsible for deciding the winning criteria and delivering the winning notification, to the relevant advertisers through a DSP. The DSP facilitates the agencies by planning and executing the ad campaigns and analyzing the best possible investments on bidding, to improve the returns on investments (ROI) of advertisers. Among the different type of ad platforms which practice RTB strategies, the main two streams are web and mobile. Thus, almost all the RTB systems facilitate both platforms in parallel without much differentiation. Consequently, in this paper, we have used the term “application” as a common term to address any mobile applications or web sites that are incorporated with an ad platform. Before describing the details, we define several other common terms which are important in understanding the RTB ecosystem and is used throughout the paper. The advertiser target is a common term to denote advertisers’ target spend, target audience and target number of conversions, all together. The target spend is the total dollar value an advertiser can spend in buying impressions during a particular campaign. The target audience of an application can be determined based on the characteristics of users, such as age, income, ethnicity, languages, has children, gender, education, etc. The conversions are also called actions and they reflect how users interact with the advertisements, such as clicks, calls, SMS, views, etc. The campaign period is the total duration of an ad campaign. To further understand the aforementioned terms let’s consider an example. A DSP runs an ad campaign for a day targeting a Unilever Shampoo product called “Dove” towards a female audience. Here Unilever is the advertiser and Dove is the product. In this instance, Unilever will decide the target spend as \$1000, the campaign period as a day, target audience as 100% female and target number of conversions as 2000 clicks.

1.2 Problem Specification

As explained earlier, at the step 5 in the RTB ecosystem, the DSP needs to determine the best bid price which can win the impression. If the DSP is not interested in winning the impression, then the bid price is mentioned as zero in the bid response. The key problem that we address in this research is, how to determine this bid price while achieving advertiser target. In relation to the above example, DSP needs to decide the most appropriate bid prices for selected set of bid requests which enable Unilever to achieve its target, among many such bid requests from many different applications with a different target audience and different winning rates, such that maximum return is achieved. The return is the numbers of actions (clicks) by the target audience at a given target spend.

As depicted in Fig. 2, we are able to gather information about applications' historical winning bid price, target audience and conversion rates using past ad campaign data. The objective is to target, applications which have the correct target audience, highest historical conversion rate and lowest historical winning bid price.

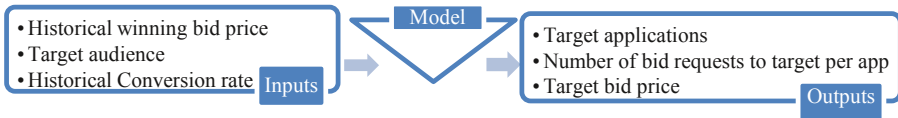


Fig. 2. Inputs and outputs

Currently, most of the DSPs carry out RTB via a greedy approach, where they try to achieve a higher number of conversions by bidding for the applications which have a relevant target audience. To increase their probability of winning, they manually decide a higher price than the known WBA (winning bid average). However, this is a very suboptimal strategy. Other prior researches focus on deciding the best bid price through a prediction algorithm [13], but as we demonstrated in this paper, due to the unpredictability and rapid variation in the RTB context, it becomes harder to predict.

1.3 Challenges

The main challenge of the research is the different dynamic aspects of the RTB process. The conventional bidding strategies completely depend on the number of bidders and their bids [4]. But in the RTB process, the dynamism exists over the number of bid requests received from each application; the different types of active applications in a particular period of time, the number of advertisers and their target spend and target goals. The target goals refer to the required number of conversions, duration of the campaign and the target audience. For example, in our dataset we identified that some mobile applications are highly active in a particular day with larger number of incoming bid requests, but in the next day, some of them didn't even appear and others had very less number of bid requests. Fig. 3 demonstrates the average number of applications that appear in DSP during different hours of the day in two adjacent weeks. As, can be seen, it is difficult to deceive any patterns out of it and so applying

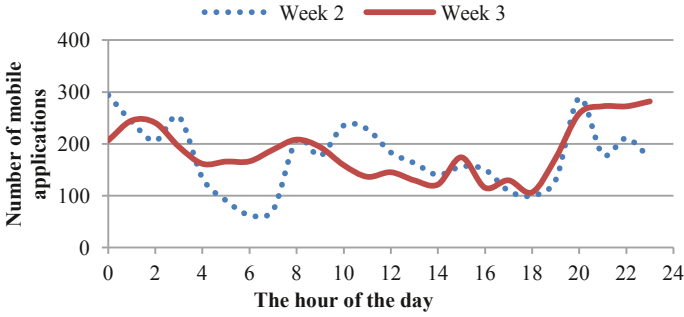


Fig. 3. The average number of distinct mobile applications which offer bid requests for DSP, in different hours of the day

any predictive logic could be futile. The difficulty of forecasting bid prices and number of impressions from the past data is elaborated in one of the following sections. Consequently, the bid price should be decided based on the advertisers’ target spend, target audience and winning rate, not like conventional settings where it is computed only based on the previous bid price.

The problem becomes additionally complex due to the following two reasons of partial data accessibility [12]. First, unlike other auction systems in the RTB process, every winning bid price is not published to all the DSPs. Therefore a particular DSP have the data related to its winning bid prices only. RTB exchanges also publish the WBA for each application. However the WBA is computed at less frequency (such as every 24 hours) and is based on a longer duration of data (such as a week), which does not add much value other than aiding the current approach of DSP bidding – bidding higher than published WBA of a desired application. Secondly, the RTB exchange doesn’t send all the bid requests to every DSP. The bid requests distribution is based on the agreement between the DSP and the RTB exchange. Both the above reasons make the global view of the RTB exchange unavailable to DSPs. As a result, we cannot develop a solution which considers the global view of the data. Any solution we design should be bound to the local view of the data i.e. from a particular DSP perspective only and will be run the DSP system.

The rest of the paper is organized as follows. The next section provides an overview for the prior researches on real time bidding strategies. Then, it explains the methodology and the dataset which is used to test the proposed models. After that each of the sections describes forecasting approach and dynamic programming approach in detail, including the problem formation. The following section describes the analysis of the dynamic programming model and its results. The last section provides the conclusion for the study.

2 Related Work

Rogers et al. [14] have proposed a probabilistic model while taking into account both the behavior of the users (advertisement viewers) and the advertisers. Their model endows a better exposure to the advertisers on the bidding strategy. The evaluation of

the model was carried out via a simulation mechanism and its objective is, to identify the most appropriate bid value for each auction and to maximize the probability of having a larger number of impressions. In other research, impressions are allocated in a randomized fashion [7]. Hegeman et al. [8] have emphasized the important criteria to build a bidding strategy based on historical value of the impression, the time or date of the impression, total allocated budget, the identity of the entity requesting, the predicted likelihood the ad will be selected, the presence of social functionality, available budget, total number of impressions of the ad, and the remaining number of impressions to be achieved etc. Besides, in other research, a bidding strategy was developed by optimizing both the budget and the bid price, which guarantees the convergence to a locally envy-free equilibrium via greedy strategy [2]. A detailed elaboration is made by Yuan et al. [16] about the RTB strategy and comprehended on how temporal behaviors, the frequency and recency of ad displays would be nontrivial. Chakraborty et al. [3] have come up with a joint optimization framework through online algorithms and stochastic model to optimize the allocation and solicitations. Their solution is an online recurrent Bayesian decision framework with bandwidth type constraints. The work carried out by Li et al. [13] is very similar to the work which we proposed in this paper, except they did not consider an advertiser's target goal as a key parameter for the model. They also tried to predict the bid value while acquiring an impression at a lowest cost. Their strategy is based on a win rate model which predicts the win rate and the winning price based on a logistic regression model and then, they derive the bidding strategies from the resulted model. As, we have discussed under the challenges, none of the aforementioned research completely admit the dynamism which is embedded in the RTB process. These past solutions do not bind with the rapid real time decision making. In addition, all these solutions try to simulate the process of the ad exchange, but there is no solution which looks at the DSP perspective which is the key business entity in the RTB ecosystem.

3 Methodology and Dataset

Initially we have evaluated the feasibility of applying forecasting approach via Autoregressive Integrated Moving Average (ARIMA) to predict the bid prices and number of bid requests. Due to dynamism in the RTB process, the accuracy of the predictions was very low. Next, we developed a dynamic programming (DP) algorithm to bid for the impressions that operates over a set of consecutive time periods which are called bid periods spread across the total campaign duration. To achieve the best outcome, the algorithm follows a model which adjusts its properties for the next bid period based on the prior period behavior. The model consists of three steps – (1) budget allocation strategy, (2) bid price adjustment strategy and (3) application allocation strategy. The model will execute all these three steps at the beginning of a particular bid period and its outcome will be applied to the campaign execution during that bid period. Such a dynamic programming approach can adapt the bidding process in the RTB successfully. The performance of the algorithm can be adjusted by reducing the duration of each bid period.

To develop a model and test it empirically, we have preserved certain huge amount of data from an ongoing RTB process of a leading mobile DSP. Compared to previous studies where the model is examined with the synthetic data, the factual data has given a proper insight into the model. The dataset includes the data for three campaigns, each of 10 days duration, run by the DSP in August 2014. It includes 6,317,443 bid requests which are spread across month of August 2014. Table 2 depicts some of the details of these three campaigns and Table 3 provides an example actual data values from our dataset for the fields listed in the Table 1.

Table 2. Data difference among three days

	Campaign X	Campaign Y	Campaign Z
Total number of bid requests	2,209,864	2,113,487	1,994,092
Total distinctive applications	240	205	160
Winning bid average for the whole campaign \$	1.06	1.15	1.08

Table 3. Actual values for the fields of bid request, bid response, win note and conversion note

Field	Actual values
Bid request	
id	9026174797775044599
timestamp	1402724400154
imp	"banner": {"top-frame":1,"id":"1","w":320,"btype":[1,4],"batrr":[3,8,9],"hmin":50,"api":[4,3,5],"wmin":300}
site/app	"id":"81134", "name":"AcademMedia Nail Manicure", "publisher":{"id":"194507","name":"AcademMedia"}, "domain":{"com.games4girls.NailManicure"}
device	"os":"Android","model":"SPH-M830"
geo	"zip":"10030","lon":-73.88476,"lat":40.73874,"city":"New York"
user	"gender":"M"
tmax	200
Bid response	
id	9026174797775044599
price	1.50
currency	USD
nurl	http://inneractive.mobilewalla.com/inneractive/win/\${AUCTION_ID}/\${AUCTION_BID_ID}/\${AUCTION_IMP_ID}/\${AUCTION_SEAT_ID}/\${AUCTION_AD_ID}/\${AUCTION_PRICE}/\${AUCTION_CURRENCY}
adid	inneractive-9026174797775044599
Win note	
id	inneractive-9026174797775044599
winPrice	0.66
currency	USD

Table 3. (Continued)

Conversion note	
impression_id	inneractive-9026174797775044599
timestamp	1402724433677

4 Forecasting Approach

In this section we verify whether the forecasting based approach can be used to determine the average winning bid price and number of incoming bid requests for applications. We segment the total campaign period in multiple bid periods. Next, to forecast the average winning bid price and the number of incoming bid requests of applications in a bid period, we fit time series to the historical values for all previous bid periods up to maximum one week.

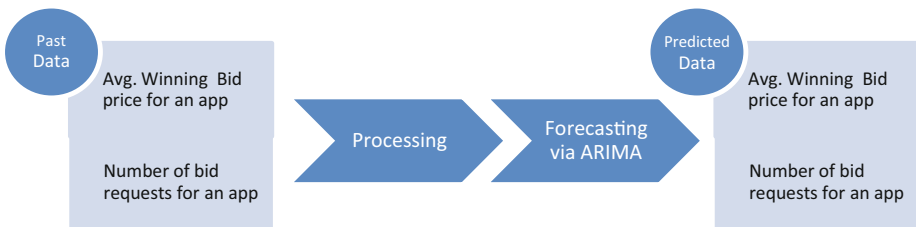
We have considered Autoregressive Integrated Moving Average (ARIMA) technique for forecasting among many time series analysis techniques. This technique is a generalized version of Autoregressive-Moving-Average (ARMA) which can only applicable with time series data. Since the average bidding price of an application is a non-stationary series, we have to apply ARIMA while taking its successive derivatives until it meets a stationary one. In this technique lags of the differenced series appearing in the forecasting equation are called autoregressive terms, lags of the forecast errors are called moving average terms, and a time series which needs to be differenced to be made stationary is said to be an integrated version of a stationary series.

A non-seasonal ARIMA model can be written as

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 e_{t-1} + \dots + \theta_q e_{t-q} + e_t$$

where y'_t is the differenced series (it may have been differenced more than once). The “predictors” on the right hand side include both lagged values of y_t and lagged errors (e_t or white noise). c is a constant. This can be further classified as an ARIMA (p,d,q) model where:

- p is the number of autoregressive terms,
- d is the number of non-seasonal differences, and
- q is the number of lagged forecast errors in the prediction equation.

**Fig. 4.** RTB forecasting strategy

Initially we have begun the analysis by identifying the order of differencing needed to stationerise the series and remove the gross features of seasonality, in conjunction with a logging which leads to variance-stabilizing transformation.

In this approach, selecting appropriate values for p , d and q is difficult. Therefore, we have used the `auto.arima()` function in R which will automatically identify the best ARIMA model based on the dataset. This function uses a variation of the Hyndman and Khandakar algorithm [10] which combines unit root tests, minimization of the corrected version of the Akaike information criterion (AIC) for ARIMA model and the Maximum Likelihood Estimator (MLE). The reason to find the AIC is because it can help to determine the order of an ARIMA model. It can be written as follows,

$$AIC = -2\log(L) + 2(p + q + k + 1)$$

Where L is the likelihood of the data with, $k = 1$ if $c \neq 0$ and $k = 0$ if $c = 0$. Moreover, the corrected AICc can be written as

$$AICc = AIC + 2(p + q + k + 1)(p + q + k + 2)T - p - q - k - 2$$

Once the model order has been identified (i.e., the values of p , d and q), we need to estimate the parameters $c, \phi_1 \dots \phi_p, \theta_1 \dots \theta_q$. To estimate them R uses MLE, because this technique finds the values of the parameters which maximize the probability of obtaining the data that we have observed.

Based on the output model, the prediction of the next bid period's average win price is carried out through the one-step forecast technique. One-step forecast can facilitate to predict the value of an endogenous variable in the current period by using the estimated coefficients, the past values of the endogenous variables, and any exogenous variables. In simple terms it can predict next period values while giving higher weight to the previous period value of the time series.

The forecasting experiment was carried out using the dataset which was described earlier with 15 minutes bid periods, i.e. bid price and number of bid request for each application are predicted at every 15 minutes based on time series fitted on past data. To generate the accurate time series data, we relied on at least one week's (seven days) of historical data. The predicted value at each bid period is compared with the actual value using few accuracy measurement techniques as described below. In each equation, t represents the bid period and n the number of total bid periods in a week.

Mean Absolute Percentage Error (MAPE): This determines the size of the error by computing the average of the unsigned percentage error [9]. The equation of the measurement is as follows:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Actual_t - Forecast_t|}{Actual_t}$$

Mean Absolute Deviation (MAD): This is also termed as Mean absolute error where accuracy is computed in the same units as the data. Similar to the MAPE, MAD also

helps to conceptualize the amount of error [9]. The calculation can be carried out using following equation:

$$MAD = \frac{\sum_{t=1}^n |Actual_t - Forecast_t|}{n}$$

Root Mean Square Error (RMSE): This can be used to measure the differences between predicted values and actually observed values. RMSE is the standard deviation of the differences between predicted and actual values [9]. Following equation depicts how it can be computed,

$$RMSE = \sqrt{\frac{\sum_t^n (Forecast_t - Actual_t)^2}{n}}$$

Table 4 exemplifies the accuracy calculation statistics for the forecasting on both bid price and number of bid requests received for each of the 4 weeks in the dataset.

Table 4. Accuracy measurement for forecasting results

	Week 1	Week 2	Week 3	Week 4
<i>Bid Price Forecast</i>				
Mean Absolute Percentage Error (MAPE)	14.55	20.51	65.78	9.02
Mean Absolute Deviation (MAD)	0.11	0.14	0.41	0.11
Root Mean Squared Error (RMSE)	0.13	0.17	0.56	0.12
<i>Bid request count forecast</i>				
Mean Absolute Percentage Error (MAPE)	34.54	24.98	78.97	24.71
Mean Absolute Deviation (MAD)	20.05	19.64	17.44	12.66
Root Mean Squared Error (RMSE)	26.88	28.78	21.49	13.72

As per the results listed in Table 4, all the three accuracy measurements have very high values which reflect that the accuracy of the forecast is very low. Additionally, there is a considerable difference between the weeks' results, for an example, week 3 has smaller accuracy values than other weeks. This also indicates the unreliability of forecasting based approach in RTB bidding.

Thus above analysis demonstrates that forecasting the number of bid requests and winning bid price based on historical values has very low accuracy. Such forecasted values can't be reliably used for bidding by DSP in RTB exchange.

Following this we propose a dynamic programming based approach to determine the bid price and the number of bid requests from the each selected applications.

5 Dynamic Programming Approach

In this section we present a dynamic programming based approach, where we determine the bid price of each application to optimize the advertiser's target click (or other

actions) with a given spending at each bid period. Before probing the details of the approach, a list of notations which are used throughout the rest of the paper is declared in Table 5.

Table 5. Notation for the model descriptions

Indices	
j	index for applications $j = 1, \dots, J$
b	index for the selected applications for bidding $b = 1, \dots, B$
x	index for applications' target audience characteristics $x = 1, \dots, X$
d	index for advertiser's desired target audience characteristics $d = 1, \dots, D$
t	index for a bid period $t = 1, \dots, T$
Parameters	
S_t	remaining budget of an advertiser at bid period t
I_{jt}	total number of impressions available for application j at bid period t
P_{jt}	bid price for application j at bid period t
C_{jt}	conversion rate for application j at bid period t
Y_{xj}	target audience options for characteristic x on application j
Y_d	advertiser's target audience option for characteristic d
J	total number of available applications to bid
X	total number of target audience characteristics
T	total number bid periods performed during an ad campaign period
B	total number of applications which have selected to bid
A_j	accessible target audience for application j
W_{bt}	target winning rate for application j at bid period t
P_{bt}	bid price for application b at bid period t
S_t	budget allocation for bid period t
M_t	moving average on total bid requests at bid period t
MAR_t	moving average ratio at bid period t
V_{bt}	number of impressions won in application b at bid period t
Decision variables	
K_{bt}	number of impressions selected to bid from application b at bid period t

The Dynamic programming model is developed to achieve advertiser target by running the ad campaign during the whole campaign period. To utilize the full campaign period properly, dynamic programming approach divides the campaign period into equal multiple intervals which are called bid periods. As explained in Fig. 2, when the campaign is running, based on the previous bid period's data, the proposed dynamic programming model will determine the next bid period's target bid price, target applications and number of bid requests per app. The model mainly consists of three steps (see Fig. 5). Firstly, the campaign budget is distributed and allocated to the next bid period, based on the remaining number of periods and prior period's number of received bid requests. Secondly, based on the bid price(s) of the selected application(s) during the previous bid period, it adjusts the bid prices to obtain the target winning

rate in the next bid period. Thirdly, the application selection strategy facilitates to identify the best possible applications which optimize the advertiser target through maximizing advertiser utility value with respect to the accessible target audience, conversion rate and the bid price. Since, these three steps are required to be executed, during each bid period; it depicts an autonomous and dynamic bidding strategy for the whole campaign. The duration of a bid period will be decided by the DSP, thus it could be on a daily, hourly, 30, 15 or 5 minutes, etc. Next we describe each of these steps.

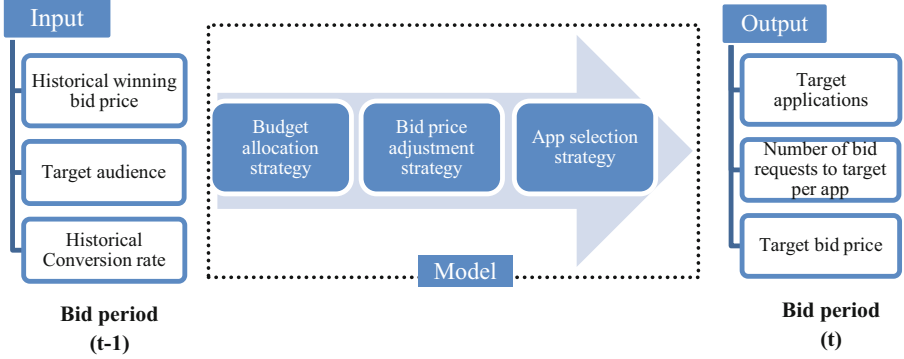


Fig. 5. Three steps of the auto pricing strategy

5.1 Budget Allocation Strategy

When a new bid period is starting, we need to allocate the target spend for each remaining bid period, depending on the previous periods' total number of bid requests received and the remaining target spend. As, real time bidding is a very rapidly changing environment; the number of bid requests which is received to a DSP can fluctuate in adjacent bid periods capriciously. To enforce this dynamism, we have calculated the moving average value of the total bid requests for each bid period until the last executed bid period.

$$M_{t-1} = \frac{\sum_{t-1} \sum_j I_{jt-1} \times t-1}{\sum_{t-1}}, \forall j \quad (1)$$

According to the Eq. (1), we have considered the index of the bid period as the weight of each period's total bid request. As a result, the current period has the highest weight compare to the previous bid periods. Using moving average value of the last period ($t - 1$) and the bid period before the last period ($t - 2$), we compute the Moving Average Ratio (MAR) for the current period as shown in Eq. (2). This will help to apply the recent changes of the received bid requests in the last bid period in the model with respect to the previous bid periods.

$$MAR_t = \frac{M_{t-1}}{M_{t-2}}, \forall t \quad (2)$$

Whilst, we have MAR for current period, then we can compute the budget for the next period as shown in Eq. (3),

$$S'_t = \frac{S_t \times MAR_t}{T}, \forall t \quad (3)$$

5.2 Bid Price Adjustment Strategy

In RTB, the winning bid price has a dynamic behavior which varies frequently. Thus the winning bid price which is predicted for a particular bid period might not be the optimal bid price in the following period. If we bid higher than the optimal bid price, we will pay more than required for the desired goal, then we would not be able to achieve a higher number of conversions due to the restricted spend; on the other hand, if we bid for a lower bid price, then also we would not win enough to achieve a higher number of conversions, due to limited duration of the campaign. As a result, in this step, we adjust the bid price with regard to the actual winning rate. The actual winning rate is defined in Eq. (4):

$$W_{bt} = \frac{V_{bt}}{K_{bt}}, K_{bt} > 0, b \in J, \forall b \quad (4)$$

In our model, the actual winning rate is calculated at the beginning of each bid period and it is used to adjust the bid price for the next period. Mainly, the bid price adjustment strategy is developed on the idea of maintaining a higher winning rate for a lower bid price. For instance, in a particular application, in a particular bid period if the numbers of actual winning bid requests are equal to the number of expected winning bid requests, it is possible to reduce the bid price. However, in many situations, the actual winning rate is lower than the expected winning rate, hence, to increase the winning rate for the subsequent bid period, we increase the last period bid price with respect to the actual winning rate for the last period. Furthermore, if the actual winning rate for the last period is zero for a particular application, then we increase its bid price for the next period by a constant value multiplication.

The new target bid price for the next bid period is computed as Eq. (5),

$$P_{bt} = \begin{cases} P_{bt-1} \times \alpha, & W_{bt-1} = 0, 1 \leq \alpha \leq 2 \\ \frac{P_{bt-1}}{W_{bt-1}}, & 1 > W_{bt-1} > 0 \\ P_{bt-1} \times \beta, & W_{bt-1} = 1, 0.5 \leq \beta \leq 1 \end{cases} \quad (5)$$

In the Eq. (5), we have defined two constants α and β , as thresholds to limit the scope of bid price adjustment. We have tested and proved their boundaries. Furthermore, as per the experimental results we have identified the best possible values for the α and β , as respectively 1.5 and 0.8. Apart from the bid price adjustment on the selected applications, the system also should keep track of all the applications and their details which generated the bid requests in the previous period. We apply the price adjustment strategy to all the targeted applications during the last bid period. Therefore, to condense the effect from applications which was not targeted in the previous period,

we have updated their total number of bid requests in the last period as 1. Even the model selects such applications to bid in the next period, it can expect only one bid request to be bid. Resultantly, if such bid requests couldn't win, then the model will adjust their bid prices for the next bid period ($t + 1$) and it will increase the probability of winning for the next period. The Eq. (6) defines this constraint. This strategy has been experimentally evaluated and the results are explained in the analysis section.

$$I_{jt} = 1 \quad j \notin B, \forall j \quad (6)$$

5.3 App Selection Strategy

Accessible Target Audience

In most cases, publishers provide the target audience for their applications to the DSPs based on user characteristics such as a percentage of males use the app, a percentage of different age groups use the app, etc. With such information and the advertiser's preferences, the following process defines the strategy to identify the accessible target audience. For example, if an advertiser requests to target his advertising campaign towards female users, then the best impressions to be published for such advertisements, are those, which belong to an application with target audience gender as 100% female and the worst case scenario is an application with 100% male target audience. In a similar way, if there is an advertiser who requires targeting his ad campaign for females who are aged under 30. Also, if there is a particular application with 60% female and 80% aged under 30 years as target audience, then accessible target audiences will be computed by the product of female and age group percentages (48%). Hence, the accessible target audience for a particular application is captured based on the product of relevant advertiser's target audience characteristics and it can be defined as Eq. (7),

$$A_j = \prod_x Y_{xj}, \quad Y_{xj} \in Y'_d, x \in d, \forall j \quad (7)$$

Optimization Strategy

In the RTB process, the advertiser is looking for higher returns, through a higher number of conversions for his investments. Therefore, the goal of the model is to increase the advertisers' returns on investments (ROI) of buying impressions, by optimizing their target in all the available listed applications at a certain bid period. In the process of establishing the model, we can define a utility value to demonstrate the effectiveness of achieving conversions with regard to the advertiser target. The utility value can be determined based on the return rate. By multiplying conversion rate and accessible target audience, we can calculate how many conversions of the required target audience can be obtained for a particular bid value of an impression. Then, to identify the exposure or return per dollar, the return rate can be estimated as Eq. (8),

$$\text{Return rate } RR_{jt} = C_{jt-1} A_j / P_{jt}, \quad \forall j, t \quad (8)$$

Since we have the return rate per impression, the value of an effective number of conversions per dollar can also be computed. This value reflects the utility value for an advertiser. Eq. (9), defines the utility value for the advertiser on a particular application j .

$$\text{Utility value } U_{jt} = \frac{C_{jt-1} A_j K_{jt}}{P_{jt}}, \quad \forall j, t \quad (9)$$

When maximizing the utility value we can increase the return rate and effective number of conversions that can be attained for a lower bid price. In simple terms, when maximizing the utility value we can access to higher conversion rate, higher accessible target audience, lower bidding price and higher number of selected impressions. However, when identifying the highest utility value among all the applications' utility values, we have to endure following constraints of the advertiser.

The first constraint, Eq. (10), is the number of impressions chosen from a particular application is limited to its total number of available impressions. As per the results of earlier section, we consider predicted value of the number of available impressions for the next bid period. Therefore, when selecting impressions to bid, the total number of impressions which can be selected is limited to the number of available impressions of that particular bid period. This should be true for all the listed applications.

$$K_{jt} \leq I_{jt}, \quad \forall j, t \quad (10)$$

The next constraint, Eq. (11) is, when selecting applications, the spending for all the impressions should be less than or equal to the target spend for the period. The mathematical formulation for this constraint can be defined as follows:

$$S'_t \geq \sum_j P_{jt} K_{jt}, \quad \forall t \quad (11)$$

However, since impressions are selected only from the applications which have maximum utility value, for the remaining applications, the number of selected impressions could be zero. Therefore, the constraint, Eq. (12) should hold,

$$K_{jt} \in \mathbb{Z}^*, \quad \forall j, t \quad (12)$$

Based on the above three constraints, the utility values can be maximized to find the most suitable applications which could optimize the advertiser expectations. As a fact, we can define the Eq. (13) to maximize the sum of the utility value across all applications at a particular bid period (t) while bonding through the constraints.

$$\text{Maximize } \sum_j U_{jt} \rightarrow \text{Maximize } \sum_j \frac{C_{jt-1} A_j K_{jt}}{P_{jt}}, \quad \forall t \quad (13)$$

According to the value of K_j , we can find the actual number of impressions to bid for each application. The applications whose impressions are not selected will not be considered for the bid price adjustment strategy of the next bid period.

6 Analysis and Results

As we discussed in the prior section, the problem has been formulated using mathematical modeling, and we have developed a DP algorithm [1] to test the model. Since the model is developed in such a way that, each bid period's inputs are set based on the previous bid period's output, this can be implemented incrementally while adjusting the bid prices and target spend dynamically during the algorithm execution. The algorithm was coded using Java programming language and solution to the optimization problem is implemented using existing free and open-source Java library called Java Optimization Modeler (JOM). It offers a full pledged platform to model Java programs and solve optimization problems.

To evaluate the efficiency and effectiveness of the model, the metric, i.e. Target Spend per Conversion (TSPC), as Eq. (14) has been defined,

$$TSPC = \frac{\text{Target spend per campaign}}{\text{Total number of conversions}} \quad (14)$$

Align with formulated metric in the above; the objective of our analysis is to demonstrate that, the proposed model can accomplish lower TSPC while achieving higher number of conversions. According to Table 6, TSPC values for each different target spends increase, when the bid period is increased. To further understand this scenario, we have demonstrated it in Fig. 6. Comparing Fig. 6, (a) and (b), we can determine that, when the bid period is small (this means that higher number of bid periods), we can achieve a higher number of bids for a lower TSPC value.

Table 6. Model behavior at different bid periods

Campaign X		5 minutes		15 minutes		30 minutes		60 minutes	
		TSP C	conver- sions	TSP C	conver- sions	TSP C	conver- sions	TSPC	conver- sions
Tar- get spend	\$ 1000	0.14	6621	0.21	4555	0.28	3879	0.46	1988
	\$ 2000	0.19	8961	0.23	6878	0.31	5006	0.51	2731
	\$ 3000	0.22	9877	0.27	7543	0.33	5766	0.54	3104

However, when a particular bid period is considered, TSPC increases with respect to the target spend. That is because, when we allocate a higher budget for a campaign, the allocation for a particular period is also high. Since, the model has a lower granularity to select applications; it will also bid for the applications with higher bid prices to maximize the number of conversions.

As per the above scenario, we can demonstrate that, when the bid period is less, the model accuracy will be increased. Nevertheless, when the target spend is high, it will try to achieve more conversions from the applications with the higher bid prices. The insight from this analysis is, *when an advertising campaign is running in RTB, it is needed to maintain the bid period at a minimum level.*

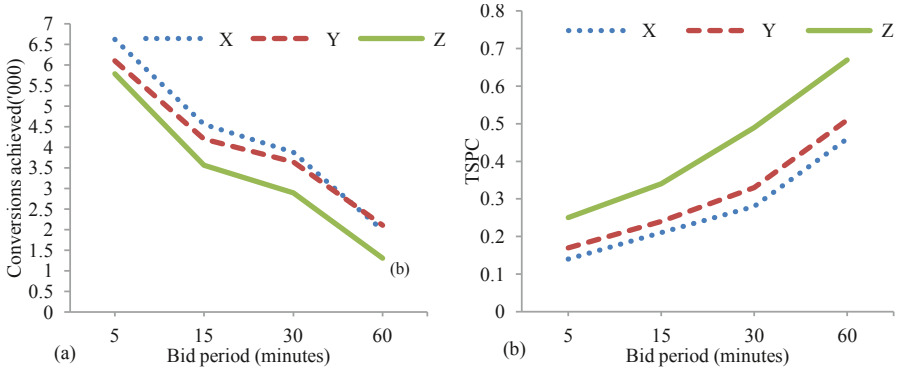


Fig. 6. (a) Conversions achieved vs. Bid periods and (b) TSPC vs. Bid periods in X, Y and Z campaigns

To validate the model with Eq. (6), we primarily look at how the model behavior would change based on the number of bid requests of the inactive applications, during the previous bid period. Resultantly, we test the model based on their last updated bid requests and updating them to single bid request and 100 bid requests. The analysis is performed for three campaigns when the bid period is 5 minutes. The results are listed in Table 7.

Table 7. Model behavior based on the number of bid requests of the applications which are not active during the previous period

	Based on last updated bid requests		Based on single bid request		Based on 100 bid requests	
	TSPC	# conversions	TSPC	# conversions	TSPC	# conversions
Campaign X	0.18	4883	0.14	6621	0.19	4412
Campaign Y	0.21	4411	0.17	6103	0.28	3631
Campaign Z	0.27	3867	0.25	5789	0.32	3347

As depicted in Table 7, we can get a better outcome, when the number of bid requests is adjusted to 1 for all the inactive applications during the previous period. Moreover, this provides added advantage, if the applications with a single bid request are selected by the optimization strategy, then based on their winning outcomes, the model can further adjust their bid prices and optimize the winning outcome for the next bid period. In parallel, when the bid price is increased, by 1.5 times for an application which couldn't win any impressions during the previous period the model provides two advantages. The first advantage is, its return rate will be reduced according to Eq. (13). This will increase the probability of bidding for another application with a higher return rate. Secondly, if it is still selected by the optimization strategy, then

there is a high likelihood to win its impressions due to the high bid price. Insight of the analysis is to understand two techniques which help to increase the model performance; the first technique is, *assign the number of bid requests to 1 for all the inactive applications in the previous bid period* and the second technique is, *increase the bid price for all the unsuccessful applications in the previous period*.

7 Conclusion

In conclusion, we have made a few clear contributions in this study. First, we explained the problem of bid price determination from the DSP perspective in a RTB system. Second, we demonstrated that due to inherent dynamism in the RTB ecosystem, the forecasting based approach to determine the bid price based on historical WBA values will not work. Third, we presented a dynamic programming based approach to adjust the bid price of applications. We demonstrated how the parameters such as bid period and spend can affect the performance of applications. The proposed dynamic programming approach addresses the dynamism inherited in RTB process with a novel and effective solution while embedding autonomous bidding decisions into the RTB in advertising.

References

1. Bertsekas, D.P., Bertsekas, D.P.: Dynamic programming and optimal control, vol. 1(2). Athena Scientific, Belmont (1995)
2. Chaitanya, N., Narahari, Y.: Optimal equilibrium bidding strategies for budget constrained bidders in sponsored search auctions. *Operational Research* 12(3), 317–343 (2012)
3. Chakraborty, T., Even-Dar, E., Guha, S., Mansour, Y., Muthukrishnan, S.: Selective call out and real time bidding. In: Saberi, A. (ed.) WINE 2010. LNCS, vol. 6484, pp. 145–157. Springer, Heidelberg (2010)
4. Cui, X., Lai, V.S.: Bidding strategies in online single-unit auctions: Their impact and satisfaction. *Information & Management* 50(6), 314–321 (2013)
5. Econsultancy 2013. Online Advertising Survey (2013), <https://econsultancy.com/reports/online-advertising-survey/>
6. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. National Bureau of Economic Research (2005)
7. Ghosh, A., McAfee, P., Papineni, K., Vassilvitskii, S.: Bidding for representative allocations for display advertising. In: Leonardi, S. (ed.) WINE 2009. LNCS, vol. 5929, pp. 208–219. Springer, Heidelberg (2009)
8. Hegeman, J., Yan, R., Badros, G.J.: Budget-based advertisement bidding, US Patent US20130124308 A1 (2011)
9. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. *International Journal of Forecasting* 22(4), 679–688 (2006)
10. Hyndman, R.J., Khandakar, Y.: Automatic time series for forecasting: the forecast package for R (No. 6/07). Monash University, Department of Econometrics and Business Statistics (2007)

11. IAB 2014, Openrtb api specification version 2.2., http://www.iab.net/media/file/OpenRTBAPISpecificationVersion2_2.pdf
12. King, M., Mercer, A.: Problems in determining bidding strategies. *Journal of the Operational Research Society*, 915–923 (1985)
13. Li, X., Guan, D.: Programmatic Buying Bidding Strategies with Win Rate and Winning Price Estimation in Real Time Mobile Advertising. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014, Part I. LNCS, vol. 8443, pp. 447–460. Springer, Heidelberg (2014)
14. Rogers, A., David, E., Payne, T.R., Jennings, N.R.: An advanced bidding agent for advertisement selection on public displays. In: *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, p. 51. ACM (2007)
15. Yahalom, et al.: Bidding for impressions, U.S. Patent Application 13/282,489 (2011)
16. Yuan, S., Wang, J., Zhao, X.: Real-time bidding for online advertising: measurement and analysis. In: *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*, p. 3. ACM (2013)