Advertising in Google Search

Deriving a bidding strategy
in the biggest auction on earth.

Anne Buijsrogge
Anton Dijkstra
Luuk Frankena
Inge Tensen

Bachelor Thesis
Stochastic Operations Research
Applied Mathematics

University of Twente

June 18, 2012

Contents

Abstract

This research aims to derive an optimal bidding strategy for keyword auctions in Google Search from the perspective of the advertiser. It presents a model for the expected profit per view, depending on either the bid or the obtained position of the advertiser. In a subsequent analysis, the position and bid are optimized to reach the maximum profit per view. In the model the AdRanks are assumed to be distributed by a general probability distribution. The analysis is restricted to the uniform, exponential and normal distributions. In addition an approximation of the expected profit per view is derived and subsequently tested by Monte Carlo simulation for different distributions and parameters. Finally, this research presents an algorithm to determine the AdRanks of the other advertisers, which may be used for vindictive bidding.

1 Preface

We have conducted this research on Google auctions for our bachelor thesis of Applied Mathematics at the University of Twente. The project was commissioned by Dennis Doubovski, owner of Social Mining, a company that specializes in optimization of online advertising campaigns. Social Mining is searching for better ways of advertising in Google AdWords. Our internal supervisors were Nelly Litvak and Judith Timmer of the chair Stochastic Operations Research. This cooperation has resulted in this report.

2 Introduction

The market of online advertising is huge and still growing. Google has the biggest market share in online advertising. When using Google Search, advertisements are shown. The order in which they are displayed is determined by an online auction. In figure (1) a heat map of Google Search is shown, displaying where users look and click when their search results are shown. This heat map demonstrates that most people look and click at the advertisements at the top of the page.

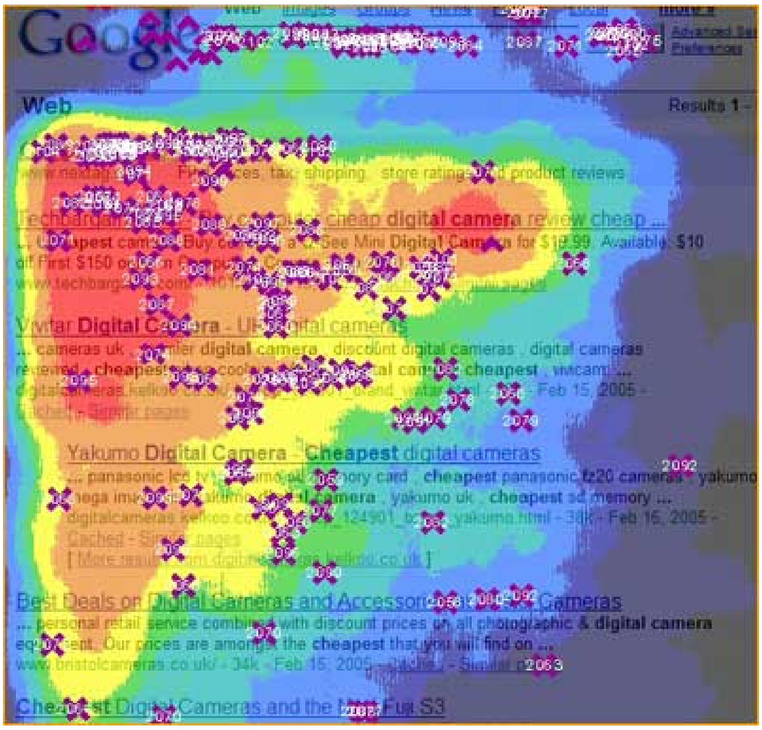


Figure 1: Heat Map. The crosses are clicks and the red area is the region most looked at.

Companies can create advertisements in Google AdWords, which are displayed in Google Search. In Google AdWords the advertisers select the keywords for their advertisements, the maximum price per click they are willing to pay and the budget for a specific keyword. The position the advertiser gets depends on the money they want to spend and the quality of the advertisement. In the next subsection the mechanism of this auction of Google Search is explained in more detail.

Generalized Second Price Auction

Each time a user searches on a keyword in Google Search, advertisements are shown by Google. The position of these advertisements is determined by an online auction, known as Generalized Second Price (GSP) auction. In this auction the bidders each place a bid, unknown to the other bidders. The bidder with the highest bid wins the auction and pays the price of the bid just below his bid, that is why it is called a second price auction.
In the GSP auction used by Google the position of an advertiser is determined by the AdRank, this is the quality of the advertisement multiplied by the maximum amount the advertiser wants to pay per click. This means that the advertiser with the highest bid does not always get allocated the best position, but the best position goes to the advertiser with the highest AdRank. These advertisement positions are called slots, with the first slot as the highest position.
Each time a user of Google Search clicks on an advertisement, the advertiser has to pay an amount of money. This is called the cost per click (CPC). According to the idea of the Generalized Second Price auction, the CPC paid by an advertiser X is the AdRank of the advertiser just below X divided by the quality of the advertisement of X.
The number of times users click on an advertisement depends on the position in which the advertisement is shown. The click through rate (CTR) is defined as the number of clicks on the advertisement divided by the number of views. As figure (1) shows, the CTR is higher on top of the page. The total price an advertiser has to pay depends on the CTR and the CPC. It is very difficult, however, to know the exact value of the CPC, because it remains a well-kept secret how Google computes the quality of an advertisement exactly.

2.1 Contributions

The first contribution is giving insight into how an advertiser who is using Google AdWords should bid in order to get a maximum profit per view. This is done with the help of a model. Based on the distribution of bids of the other advertisers the model determines what would be the optimal bid and optimal position to reach the maximum profit per view. To do so, some different distributions for the AdRanks are examined. This leads to analytical equations that give insight into a bidding strategy. Real-world data are used to get meaningful parameters. The second contribution is making a simulation of an auction in Google Search. Finally, a contribution is made by describing an algorithm which determines the AdRanks of the other advertisers.

2.2 Related Work

Generalized Second Price auctions have been studied by Easley and Kleinberg in [1]. In [1, Ch.9] Easley and Kleinberg explain which auction types are common and how they work. In [1, Ch.15] Easley and Kleinberg explain how sponsored search markets work and discuss the principle of truthful bidding. This principle shows that the best bidding strategy is to make a bid that equals the price one is willing to pay, bidding higher or lower than this bid may have a negative effect on the profit.
The principle of GSP auctions is further examined in [2] by Pin and Key. They have proposed a model to predict the number of clicks received by the advertisers. In [3] Zhou and Lukose describe a strategy called vindictive bidding, in which advertisers bid just below the bid of the advertiser above them, to make the price for the other advertiser as high as possible. This will be discussed in section 7.
In [4] Feldman, Muthukrishnan, Pál and Stein describe how to distribute the advertiser's budget among the search queries. In [5] Kitts and Leblanc describe a bidding strategy for Overture, a programme like AdWords for Yahoo. In [6] Feldman, Meir and Tennenholtz propose a slight modification to the GSP mechanism. This leads to a higher revenue for the seller of the auction, which in this case is Google. In [7] Aggarwal, Feldman and Muthukrishman study prefix position auctions where advertiser i can specify that he is interested only in the top κ_i positions. This is a variation to the GSP mechanism. In [8] Garg and Narahari describe a new mechanism to model the sponsored search auction. They compare it to the GSP and the Vickrey-Clarke-Groves mechanism, which is based on truthfull bidding, by computing the expected revenue for the search engine.
The majority of the research in online auctions is from the perspective of Google. The aim of this report is to look at the problem from the advertisers' prospective and consider the problem of optimal bidding in a simple auction so that the profit resulting from this auction is maximized.

3 Analytical Model

This section describes an analytical model for the bidding strategy in Google AdWords of a particular company that wishes to display an advertisement in Google Search.

3.1 Notations and Assumptions

The following variables are used in the model.

Variable	Description	Dim	Range
s	Number of slots	1	$s > 0$
a	Number of other advertisers	1	$a > 0$
m	Minimum price an advertiser pays for a slot	1	$m > 0$
B_{comp}	Bid of the company	1	$B_{comp} \geq m$
q_{comp}	Quality of the advertisement of the company	1	$q_{comp} \in (0, 10]$
R_{comp}	AdRank of the advertisement of the company	1	$R_{comp} > 0$
$\mathbf{r}$	AdRanks of other advertisers	a x 1	$r_i > 0 \; \forall i \in \{1, 2, ..., a\}$
$\boldsymbol{\rho}$	Probability a user clicks on a slot	a x 1	$\rho_i \in [0, 1] \; \forall i \in \{1, 2, ..., s\}$, $\rho_i = 0$ for $i > s$
$\mathbf{r}^*$	Sorted descending vector $\mathbf{r}$	a x 1	$r_i^* > 0 \; \forall i \in \{1, 2, ..., a\}$
Y	Position of the advertisement of the company	1	$Y \in \{1, 2, ..., s\}$
RPC_{comp}	Revenue per click of the company	1	$RPC_{comp} \geq 0$
PPV_{comp}	Profit per view of the company	1	$PPV_{comp} \geq 0$
CPC_{comp}	Cost per click of the company	1	$CPC_{comp} \geq m$
CTR	Click trough rate	1	$CTR \in [0, 1]$

A few assumptions were made while making this model.

- The AdRanks of two advertisers are never equal.
- The number of slots never exceeds the number of advertisers.

After the introduction of these variables it is time to start with the framework of the model.

3.2 Framework of the Model

The most important aspect of the model are the AdRanks. To get the most realistic results, all the AdRanks need to be known. As explained before, the AdRank is the bid multiplied by the quality.

$$R_{comp} = B_{comp} q_{comp}$$

If the AdRanks are known, it is possible to determine the place of the company amongst the other advertisers. To do so, the sorted vector $\mathbf{r}^*$ has to be compared with R_{comp}, resulting in the following:

$$Y = \begin{cases} 1 & \text{if } R_{comp} > r^*(1) \\ 2 & \text{if } r^*(1) > R_{comp} > r^*(2) \\ ... & \\ s & \text{if } r^*(s-1) > R_{comp} > r^*(s) \\ ... & \\ a+1 & \text{if } r^*(a) > R_{comp} \end{cases} .$$

Once the position Y is known the cost per click, CPC_{comp}, can be calculated. This is the AdRank of the advertiser just beneath the company divided by the quality of the company. Since the number of slots never exceeds the number of advertisers, Y can vary between 1 and $a+1$. Of course the company does not have to pay anything when their position is lower than s, because they do not get a slot. In the model this implies that if $a > s$ $\rho(j)$, the probability a user clicks on the slot, is defined to be zero for $j > s$. If the number of advertisers is equal to the number of slots, it is possible that the company gets the lowest slot. In this case the company has to pay the minimum price m. In formula:

$$CPC_{comp} = \begin{cases} \frac{r^*(Y)}{q_{comp}} & \text{if } Y < s \\ m & \text{if } Y = s \\ 0 & \text{if } Y > s \end{cases} . \tag{3.1}$$

The goal is to maximize the profit per view, PPV_{comp}. When a user clicks on the advertisement of the company, the company pays CPC_{comp}. If the visit of the user results in a purchase, they get revenue. The average revenue gained by the company when people click on the advertisement is called the revenue per click: RPC_{comp}. The difference between RPC_{comp} and CPC_{comp} is the profit realized by the company when a user clicks on the advertisement. On the other hand, in case the user does not click on the advertisement, the company does not pay or receive any money. The profit in case the user clicks on the advertisement has to be multiplied by the click through rate: CTR. So the formula for PPV_{comp} is as follows:

$$PPV_{comp} = (RPC_{comp} - CPC_{comp})CTR.$$

In order to calculate the maximum profit, the expectation of PPV_{comp} has to be calculated. In the next subsection an equation for the position is derived at which the maximum profit per view occurs, this is called the optimal position. After that another look will be taken at the maximum $\mathbb{E}[PPV_{comp}]$ by writing down $\mathbb{E}[PPV_{comp}]$ as a function of B_{comp} and calculating the bid where the maximum profit per view occurs, the optimal bid.

3.3 Optimal Position

First, the expected profit per view is conditioned on the position $Y = i$.

$$\begin{aligned}\mathbb{E}\left[PPV_{comp}|Y=i\right] &= \mathbb{E}\left[(RPC_{comp} - CPC_{comp})CTR|Y=i\right] \\ &= \left(RPC_{comp} - \mathbb{E}\left[CPC_{comp}|Y=i\right]\right)\rho(i) \qquad (3.2)\end{aligned}$$

The last equation follows because the expectation of the CTR given $Y = i$ results in the vector $\rho(i)$. To calculate $\mathbb{E}\left[CPC_{comp}|Y=i\right]$ formula (3.1) is used. For the first line, a distribution of $r^*(i)$ is needed. $r^*(i)$ is the i^{th} order statistic of $\mathbf{r}$.

$$\mathbb{E}\left[CPC_{comp}|Y=i\right] = \begin{cases} \frac{\mathbb{E}[r^*(i)]}{q_{comp}} & \text{if } i < s \\ m & \text{if } i = s \\ 0 & \text{if } i > s \end{cases} \qquad (3.3)$$

In section 5 this formula will be used to calculate the maximum profit per view. Bare in mind that when $RPC_{comp} < \mathbb{E}[CPC_{comp}|Y=i]$ the expected profit becomes negative, so the company should not participate in the auction.

3.4 Optimal Bid

In this part the expected profit will be conditioned on the bid B_{comp}.

$$\begin{aligned}&\mathbb{E}\left[PPV_{comp}|B_{comp} = b_{comp}\right] = \\ &\mathbb{E}\left[(RPC_{comp} - CPC_{comp})CTR|B_{comp} = b_{comp}\right] \qquad (3.4)\end{aligned}$$

To get more insight in the values of the AdRanks, several distributions are examined. First a general distribution of the AdRanks is implemented.

The probability for the company of getting position i can be determined as a function of the AdRank R_{comp}.

$$P(r^*(i) \leq R_{comp} \leq r^*(i-1)|B_{comp}) = P(r^*(i) \leq R_{comp}, r^*(i-1) \geq R_{comp}|B_{comp})$$

If $r^*(i-1) \geq R_{comp}$ there are $i-1$ advertisers with an AdRank greater than R_{comp}. Similarly, if $r^*(i) \leq R_{comp}$ the remaining $a-i+1$ advertisers have an AdRank between 0 and R_{comp}. If the distribution of the AdRanks is known, the probabilities of these events can be calculated. These probabilities have to be muliplied by the number of ways the $i-1$ advertisers can be chosen. This results in the following:

$$\begin{aligned}&P(r^*(i) \leq R_{comp}, r^*(i-1) \geq R_{comp}|B_{comp}) = \\ &\binom{a}{i-1}\left(p(B_{comp})\right)^{i-1}\left(1-p(B_{comp})\right)^{a-i+1}, \qquad (3.5)\end{aligned}$$

with $p(B_{comp}) = P(r(i) \geq R_{comp}|B_{comp})$.

Of course the exact $p(B_{comp})$ depends on the distribution of the AdRanks.
To calculate the expected profit per view, the full expectation formula, conditioning on $Y = i$ is used. Thus from (3.4) and (3.5) the following formula gives

the total expected profit per view.

$$\begin{aligned}
&\mathbb{E}[PPV_{comp}|B_{comp} = b_{comp}] \\
&= \sum_{i=1}^{a+1} \mathbb{E}[PPV_{comp}|B_{comp} = b_{comp}, Y = i] P(Y = i|B_{comp} = b_{comp}) \\
&= \sum_{i=1}^{a+1} \mathbb{E}[PPV_{comp}|B_{comp} = b_{comp}, Y = i] \\
&\cdot P(r^*(i) \leq R_{comp}, r^*(i-1) \geq R_{comp}|B_{comp} = b_{comp}) \\
&= \sum_{i=1}^{a+1} \binom{a}{i-1} (p(b_{comp}))^{i-1} (1 - p(b_{comp}))^{a-i+1} \\
&\cdot \mathbb{E}\left[(RPC_{comp} - CPC_{comp})CTR|B_{comp} = b_{comp}, Y = i\right] \qquad (3.6)
\end{aligned}$$

Formula (3.1) cannot be computed with the order statistics, because the knowledge of the bid gives extra information about the probabilities that the other advertisers are above or beneath the company. This is why the simplifying assumption is made that $CPC_{comp} = B_{comp}$. This is a reasonable worst-case scenario, because a company always pays less than their bid. Thus the lower boundary of PPV_{comp}, PPV^-_{comp}, is obtained in this way.

$$\begin{aligned}
&\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}] = \\
&\sum_{i=1}^{a+1} \binom{a}{i-1} (p(b_{comp}))^{i-1} (1 - p(b_{comp}))^{a-i+1} \left((RPC_{comp} - b_{comp})\,\rho(i)\right) \qquad (3.7)
\end{aligned}$$

In this formula the binomial distribution can be recognized. Hence, a short-hand notation of formula (3.7) is obtained as follows:

$$\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}] = \mathbb{E}[(RPC_{comp} - b_{comp})\rho(X+1)], \qquad (3.8)$$

with $X \sim B(a, p(b_{comp}))$.

The AdRank of the company can be changed by varying the bid of the company. The expected profit per view belonging to these bids can be calculated by (3.7) or (3.8) and a bidding strategy can be derived. As said before there is a critical treshold, namely, when $RPC_{comp} < B_{comp}$ the company should not participate in the auction, because of the negative expected profit. In section 5 various possible distributions of the AdRanks will be studied, followed by conclusions.

4 Data Analysis

In order to evaluate the parameters of the analytical model, real world data has to be used. In this section the real world data is being examined.

4.1 CTR versus Average Position

Since an advertisement with a higher position receives on average more clicks than one with a lower position, the CTR depends on the position. Recall that the CTR is the Click Through Rate, equal to the number of clicks on the advertisement divided by the number of views. This dependency between the CTR and the position is examined in this subsection. Comparable data often follow Zipf's law. The probability mass function of Zipf's law is:

$$f(k; s, N) = \frac{\frac{1}{k^s}}{\sum_{n=1}^{N} \frac{1}{n^s}}, \tag{4.1}$$

where $f(k; s, N)$ is in this case the CTR, $k = 1, .., N$ is the position, $s > 0$ is a parameter to characterize the distribution and $N > 0$ is the number of positions. The Zipf's law is observed by plotting the data on the log-log scale, because the plot then becomes close to a straight line. In order to test whether the data really follow the Zipf's law, two plots are made. Figure (2) is a plot of the average position against $f(k; 1, 8)$, so $s = 1$ and $N = 8$ in formula (4.1). Only the first eight positions have been taken into account, because for lower positions the CTR is zero most of the time. Figure (3) is the log-log plot of figure (2).

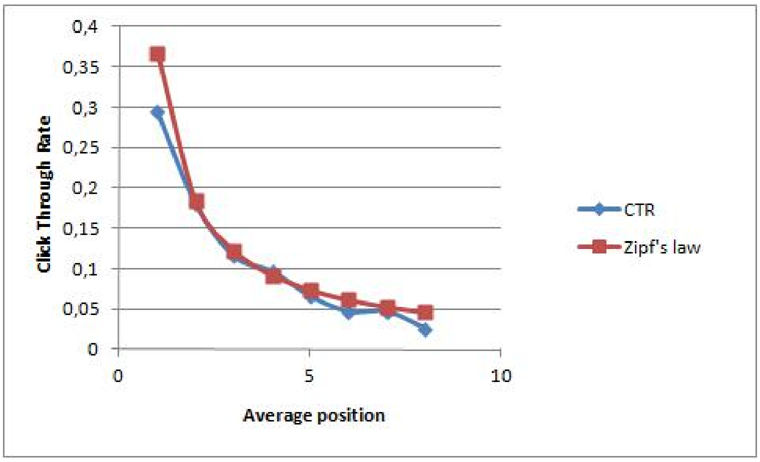


Figure 2: Zipf's law

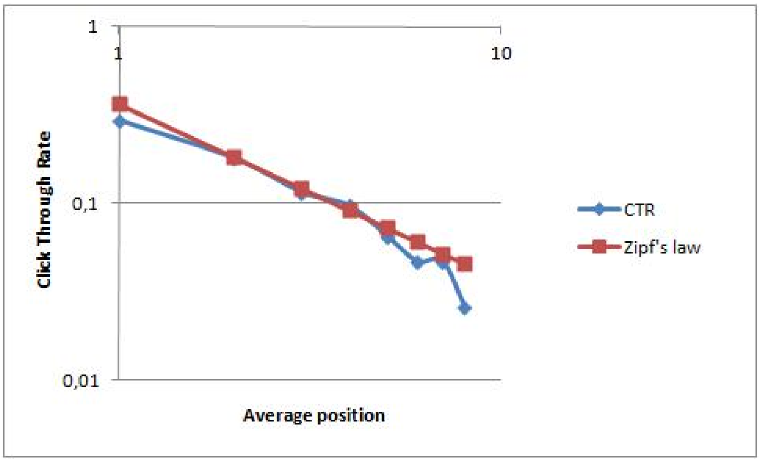
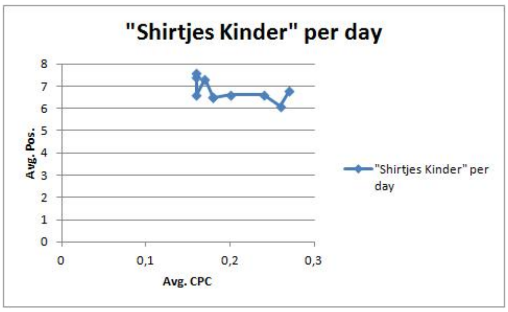


Figure 3: Zipf's law log-log

The figures show that the data approximately follows the Zipf's law. When taking $s = 2$ the average position differs more from the Zipf's law. Also for higher values of s the data do not fit better with Zipf's law. The conclusion is that Zipf's law with $s = 1$ is in this case the best approximation for the CTR, that is for $\boldsymbol{\rho}$.

4.2 Average CPC versus Average Position

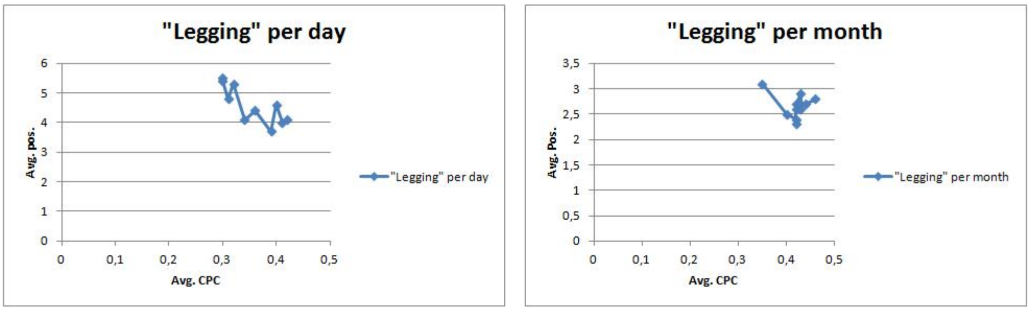
To analyse the dependency of the average CPC and the average position, several samples were taken from international campaigns. These samples have been examined per day and per month. The results are shown in the following figures. Each of the four keywords "shirtjes kinder", "legging", "babymutsjes" and "polo korte mouw" are examined in the same period of time.



(a) "Shirtjes kinder" per day

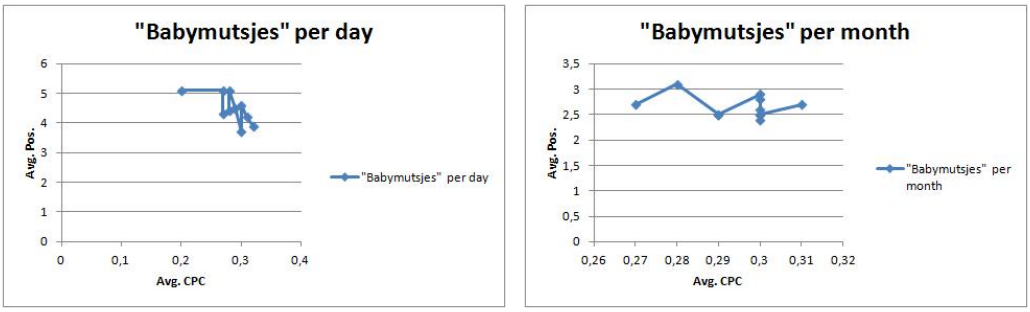
(b) "Shirtjes kinder" per month

Figure 4: "Shirtjes kinder"



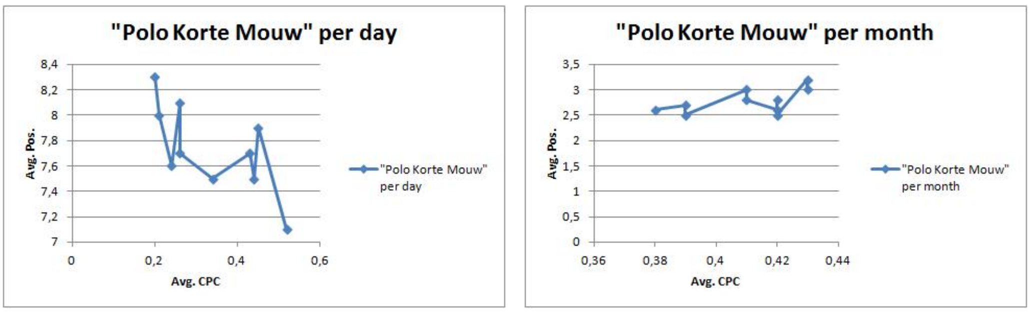
(a) "Legging" per day (b) "Legging" per month

Figure 5: "Legging"



(a) "Babymutsjes" per day (b) "Babymutsjes" per month

Figure 6: "Babymutsjes"



(a) "Polo korte mouw" per day (b) "Polo korte mouw" per month

Figure 7: "Polo korte mouw"

Figures (4a) and (4b) seem to show that for this keyword the average position is more or less equal no matter what the average CPC is. Some of the figures, like figures (5a) and (5b), seem to show that there is no influence of CPC on the position at all. Other samples seem to have a slightly upward or downward trend.
According to formula (3.1), if the CPC increases, the number of the position is decreasing, assuming that the quality stays the same. If the position number is

decreasing, the position is increasing. After all, the highest position is defined as position number one.
Why does this pattern not occur in our samples? The first explanation is that the sample size is too small. For all samples, 10 data points were used. However, creating a bigger sample size is not solving the problem completely. Secondly, in the samples the CPC does not vary sufficiently. The CPC changes at the most €0.30, that is why the position does barely vary and the results can be regarded as noise. The third and most important reason why this does not occur is because the argumentation above is based on the assumption that the bids of the other advertisers stay the same. In that case the conclusion that the company gets a higher position when they pay more is legitimate. Otherwise, when the other advertisers bid differently, the company might get a different position. Since it is not very realistic to assume that over a period of 10 months the bids of the other advertisers will not change, it is very likely that this also has influence on these results.

5 Model Analysis

In this section the model of section 3 is analysed, assuming that the AdRanks of the advertisers $1, .., a$ follow a specific probability distribution. To simplify the model, the assumption is made that the number of slots s equals $a + 1$. The analysis in this report is restricted to the uniform, exponential en normal distribution of the AdRanks.

5.1 Uniform Distribution

The assumption is made that the AdRanks of the other advertisers are uniformly distributed between α and β, with $\beta > \alpha > 0$. At first the maximum of PPV_{comp} is calculated for the optimal position. Secondly the optimal bid is examined.

Optimal Position

The probability for the company of getting a position i can be determined as a function of the AdRank R_{comp}. To calculate $\mathbb{E}[CPC_{comp}|Y = i]$ the order statistics $\mathbf{r}^*$ of the uniform distribution are needed. The expectation of CPC_{comp} is shown in the formula below.

$$\mathbb{E}[CPC_{comp}|Y = i] = \frac{\beta - \frac{i-1}{a}(\beta - \alpha)}{q_{comp}} \tag{5.1}$$

In the special case when the company is in the last slot $a+1$, $\mathbb{E}[CPC_{comp}|Y = i]$ is equal to m. If $i = a + 1$ is substituted in formula (5.1), $\mathbb{E}[CPC_{comp}|Y = i]$ becomes $\frac{\alpha}{q_{comp}}$. Now from formula (3.2) for $\mathbb{E}[PPV_{comp}|Y = i]$ it follows that:

$$\mathbb{E}[PPV_{comp}|Y = i] = \left(RPC_{comp} - \frac{\beta - \frac{i-1}{a}(\beta - \alpha)}{q_{comp}} \right) \rho(i). \tag{5.2}$$

If the assumption is made that $\rho(i)$ follows Zipf's law, formula (5.2) can be completed.

$$
\begin{aligned}
&\mathbb{E}[PPV_{comp}|Y=i] \\
&= \left(RPC_{comp} - \frac{\beta - \frac{i-1}{a}(\beta-\alpha)}{q_{comp}}\right) \frac{\frac{1}{i}}{\sum_{n=1}^{a+1}\frac{1}{n}} \\
&= \frac{1}{q_{comp}} \frac{1}{\sum_{n=1}^{a+1}\frac{1}{n}} \left(\frac{RPC_{comp}q_{comp} - \beta - \frac{\beta-\alpha}{a}}{i} + \frac{\beta-\alpha}{a}\right)
\end{aligned}
$$

This formula is decreasing in i and, since i is positive, the maximum value for $\mathbb{E}[PPV_{comp}|Y=i]$ occurs for small values of i. Hence, in this case the best strategy is to obtain the highest position under the assumption that $RPC_{comp} \geq \frac{\beta - \frac{i-1}{a}(\beta-\alpha)}{q_{comp}}$, which is equal to $\mathbb{E}[CPC_{comp}|Y=i]$.

Optimal Bid

Now the optimal bid that maximizes PPV^{-}_{comp} is computed. Since $p(B_{comp})$ is needed to substitute in formula (3.7) it has to be specified.

$$
p(B_{comp}) = \begin{cases} \frac{\beta - B_{comp}q_{comp}}{\beta-\alpha} & \text{if } \alpha \leq B_{comp}q_{comp} \leq \beta \\ 0 & \text{otherwise} \end{cases}
$$

The following formula is used to give more insight in the behaviour of $\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}]$. If the assumption is made that $\rho(i)$ follows Zipf's law, formula (3.8) becomes as follows:

$$
\begin{aligned}
&\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}] \\
&= \mathbb{E}\left[\frac{\frac{1}{X+1}}{\sum_{n=1}^{a+1}\frac{1}{n}}(RPC_{comp} - b_{comp})\right] \\
&= \frac{1}{\sum_{n=1}^{a+1}\frac{1}{n}} \mathbb{E}\left[\frac{1}{X+1}(RPC_{comp} - b_{comp})\right],
\end{aligned} \tag{5.3}
$$

with $X \sim B(a, p(b_{comp}))$.
Unfortunately, no conclusions can be drawn from this formula. This is why the analysis continues with formula (3.7).
Taking the derivative of the following formula equal to zero, the optimal bid is obtained.

$$
\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}] =
\begin{cases}
\sum_{i=1}^{a+1}\binom{a}{i-1}\left(\frac{\beta - b_{comp}q_{comp}}{\beta-\alpha}\right)^{i-1}\left(\frac{b_{comp}q_{comp}-\alpha}{\beta-\alpha}\right)^{a-i+1} & \\
\quad \cdot (RPC_{comp} - b_{comp})\frac{\frac{1}{i}}{\sum_{n=1}^{a+1}\frac{1}{n}} & \text{if } \alpha \leq b_{comp}q_{comp} \leq \beta \\
0 & \text{otherwise}
\end{cases} \tag{5.4}
$$

The derivative of the previous formula for $\alpha \leq b_{comp}q_{comp} \leq \beta$ is as follows:

$$\frac{\partial \mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}]}{\partial b_{comp}} =$$

$$\sum_{i=1}^{a+1} \left[\frac{\frac{1}{i}}{\sum_{n=1}^{a+1} \frac{1}{n}} \right] \binom{a}{i-1} \left[-q_{comp}(i-1) \left(\frac{\beta - b_{comp}q_{comp}}{\beta - \alpha} \right)^{i-2} \left(1 - \frac{\beta - b_{comp}q_{comp}}{\beta - \alpha} \right)^{a-i+1} \right.$$

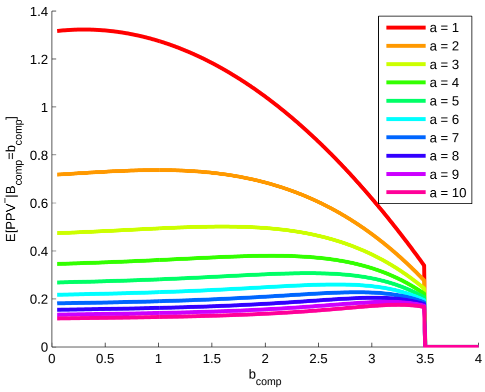
$$\cdot (RPC_{comp} - b_{comp}) + q_{comp}(a-i+1) \left(\frac{\beta - b_{comp}q_{comp}}{\beta - \alpha} \right)^{i-1} \left(1 - \frac{\beta - b_{comp}q_{comp}}{\beta - \alpha} \right)^{a-i}$$

$$\left. \cdot (RPC_{comp} - b_{comp}) - \left(\frac{\beta - b_{comp}q_{comp}}{\beta - \alpha} \right)^{i-1} \left(1 - \frac{\beta - b_{comp}q_{comp}}{\beta - \alpha} \right)^{a-i+1} \right],$$

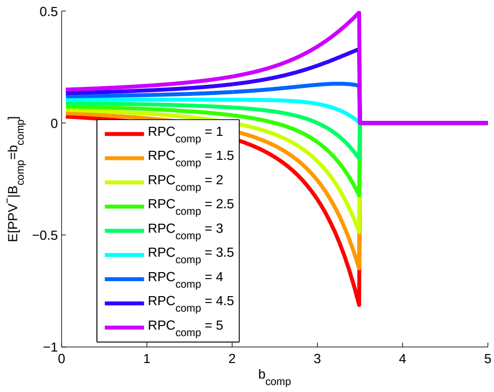
otherwise it is zero.
The roots of this derivative cannot be found analytically, that is why the maximum $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}]$ in formula (5.4) is determined numerically. The following parameters are used:

$$\begin{aligned} a &= 10, \\ RPC_{comp} &= 4, \\ q_{comp} &= 1, \\ \alpha &= 0.05, \\ \beta &= 3.5. \end{aligned}$$

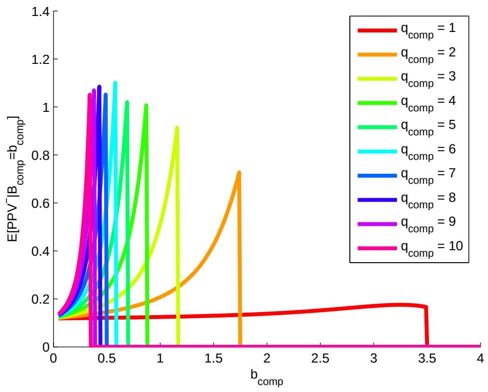
In each of the following figures one of the parameters varies.



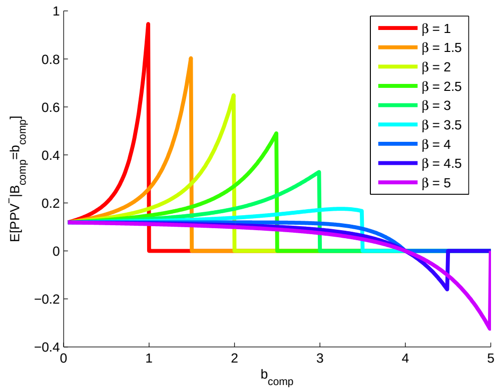
(a) Number of other advertisers a



(b) Revenue per click RPC_{comp}



(c) Quality q_{comp}



(d) Right boundary of uniform distribution β

Figure 8: Variation of the parameters for uniformly distributed AdRanks

These figures show that $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}] = 0$ if $b_{comp} \geq \frac{\beta}{q_{comp}}$, since $p(b_{comp}) = 0$ in this case.

Figure (8a) shows that with an increasing number of advertisers, the expected profit per view decreases. The explanation for this could be that if less advertisers participate in an auction, it is more likely that the company obtains a higher position, resulting in a higher click through rate. If there is only one other advertiser, the company is already certain that they will get either slot 1 or 2, so it is not necessary to place a high bid. If the number of advertisers increases, there are more competitors for the top position and the company has to make a higher bid in order to maximize their profit.

Figure (8b) displays that varying RPC_{comp} makes a big difference. If RPC_{comp} is lower then CPC_{comp}, which is equal to b_{comp}, the expected profit per view becomes negative. For example, if RPC_{comp} is 3 the expected profit per view stays positive for bids lower than 3 and becomes negative for higher bids. So the best strategy is to bid as high is possible if RPC_{comp} is bigger than b_{comp}, otherwise it is optimal to bid the minimum bid.

In figure (8c) q_{comp} is varied. Because the expected profit per view is zero in case that $b_{comp} \geq \frac{\beta}{q_{comp}}$, the value of b_{comp} for which $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}]$ becomes zero depends on q_{comp}. If q_{comp} increases, the company can make a lower bid to receive a high position, so the optimal bid will be lower.

Also in figure (8d) the expected profit per view becomes zero if $b_{comp} \geq \frac{\beta}{q_{comp}}$ and in this case this depends on β. For values of β higher than 4, the expected profit per view becomes negative because RPC_{comp} is equal to 4. For values of β lower than RPC_{comp} the conclusion can be drawn that the optimal bid is β, so it is best to bid as high as possible in this case. If this is not the case then it is optimal to bid the minimum bid.

Recall that the conclusion from the optimal position was that it is optimal to get the first position, so to bid as high as possible when $RPC_{comp} \geq \frac{\beta - \frac{i-1}{a}(\beta - \alpha)}{q_{comp}}$. This follows the conclusion for the optimal bid.

5.2 Exponential Distribution

Now the AdRanks of the other advertisers are assumed to be exponentially distributed with mean λ. Like in the case of the uniform distribution a distinction is made between the optimal position and the optimal bid.

Optimal Position

To determine the optimal position, the expected profit per view will be conditioned on the position $Y = i$ with formula (3.2). To do so $\mathbb{E}[CPC_{comp}|Y = i]$ of an exponential distribution has to be computed. This can be done with (3.3) and the expected order statistics of the exponential distribution from [9]. This results in:

$$\mathbb{E}\left[CPC_{comp}|Y = i\right] = \begin{cases} \frac{\sum_{k=1}^{a-i+1} \frac{\lambda}{a-k+1}}{q_{comp}} & \text{if } i < a+1 \\ m & \text{if } i = a+1 \end{cases} . \tag{5.5}$$

Because the summation is too difficult to substitute, boundaries of the sum of (5.5) can be derived. The sum is bounded by two integrals:

$$\int_1^{a-i+2} \frac{1}{a-t+2} dt \leq \sum_{k=1}^{a-i+1} \frac{1}{a-k+1} \leq \int_1^{a-i+2} \frac{1}{a-t+1} dt. \tag{5.6}$$

The integrals of the previous equation can be easily computed.

$$\begin{aligned} \int_1^{a-i+2} \frac{1}{a-t+2} dt &= ln(a+1) - ln(i) \\ \int_1^{a-i+2} \frac{1}{a-t+1} dt &= ln(a) - ln(i-1) \end{aligned}$$

First the boundaries are substituted in (5.5) and then the resulting expression is substituted in (3.2). The lower boundary is substituted in formula (3.2) first and results in the following formula.

$$\mathbb{E}[PPV_{comp}|Y = i] = \left(RPC_{comp} - \frac{\lambda(\ln(a+1) - \ln(i))}{q_{comp}} \right) \frac{\frac{1}{i}}{\sum_{n=1}^{a+1} \frac{1}{n}} \tag{5.7}$$

In this formula the Zipf's law for $\rho(i)$ is also substituted. Note that, since the lower boundary is subtracted from the RPC_{comp}, it now becomes an upper boundary for $\mathbb{E}[PPV_{comp}|Y = i]$. To find the optimum of this formula, the derivative is computed and set equal to zero:

$$\frac{\partial \mathbb{E}[PPV_{comp}|Y = i]}{\partial i} = \frac{\lambda - RPC_{comp} q_{comp} + \lambda ln(a+1) - \lambda ln(i)}{q_{comp} i^2 \sum_{n=1}^{a+1} \frac{1}{n}} = 0. \tag{5.8}$$

Thus the optimal position is the solution for i of the following equation:

$$\lambda - RPC_{comp} q_{comp} + \lambda ln(a+1) - \lambda ln(i) = 0. \tag{5.9}$$

Note that the denominater of formula (5.8) is never equal to zero. Solving (5.9) for i this becomes:

$$i = e^{1 - \frac{RPC_{comp} q_{comp}}{\lambda} + ln(a-1)}.$$

This is the optimal position for the upper boundary of $\mathbb{E}[PPV_{comp}|Y = i]$. This does not always has to be an integer, but then $\lfloor i \rfloor$ and $\lceil i \rceil$ have to be calculated to see which of the two has the highest $\mathbb{E}[PPV_{comp}|Y = i]$. This is probably a good approximation for the optimal position.

Secondly, the upper boundary from formula (5.5) is substituted. Also this boundary is substituted in formula (3.2) and the derivative is set equal to zero. Note that this will result in a lower boundary for $\mathbb{E}[PPV_{comp}|Y = i]$.

$$\mathbb{E}[PPV_{comp}|Y = i] = \left(RPC_{comp} - \frac{\lambda(\ln(a) - \ln(i-1))}{q_{comp}} \right) \frac{\frac{1}{i}}{\sum_{n=1}^{a+1} \frac{1}{n}}$$

The derivative is as follows:

$$\frac{\partial \mathbb{E}[PPV_{comp}|Y = i]}{\partial i} =$$
$$\frac{\lambda}{q_{comp} \sum_{n=1}^{a+1} \frac{1}{n}} \frac{\left(-\frac{RPC_{comp}}{\lambda} q_{comp} + ln(a) - ln(i-1) \right)(i-1) + i}{i^2(i-1)}.$$

So the optimal position is the solution of the following formula:

$$\left(-\frac{RPC_{comp} q_{comp}}{\lambda} + ln(a) - ln(i-1) \right)(i-1) + i = 0.$$

This equation is not solvable analytically, so the boundaries as well as the original formula are implemented numerically. In the following figure these parameters are used:

$$\begin{aligned} a &= 10, \\ RPC_{comp} &= 4, \\ q_{comp} &= 1, \\ \lambda &= 1.75. \end{aligned}$$

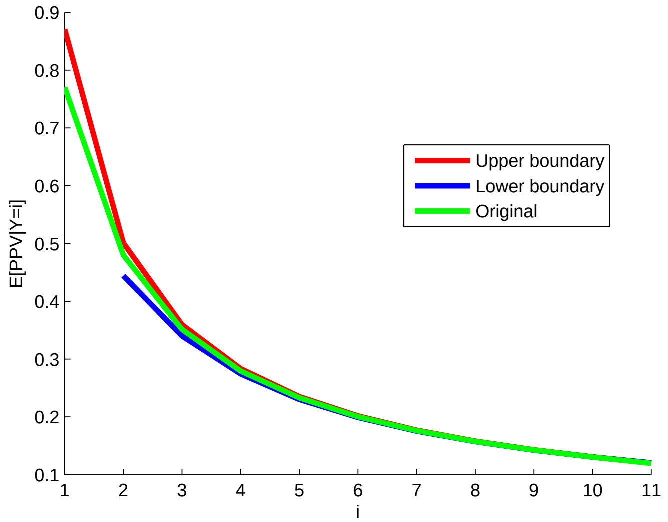


Figure 9: $\mathbb{E}[PPV_{comp}|Y=i]$ with the boundaries

The figure shows that the upper and lower boundaries result in a good approximation for the original problem and it shows that with these parameters the first position is the best one. Next, the optimal bid will be analysed and later on it will be compared to the optimal position.

Optimal Bid

Now the maximum PPV_{comp}^{-} is calculated for the optimal bid. For the exponential distribution the probability $p(B_{comp})$ in formula (3.7) becomes:

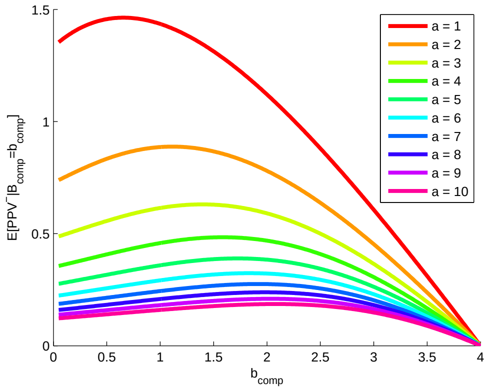
$$p(B_{comp}) \quad = \quad e^{-\frac{1}{\lambda}B_{comp}q_{comp}}.$$

With this probability the maximum PPV_{comp}^{-} can be calculated in the same way as in the previous subsection. Again a binomial distribution can be recognized, the only difference is $p(b_{comp})$. So (5.3) holds true for the exponential distribution as well. The derivative of PPV_{comp}^{-} is also taken in this case.

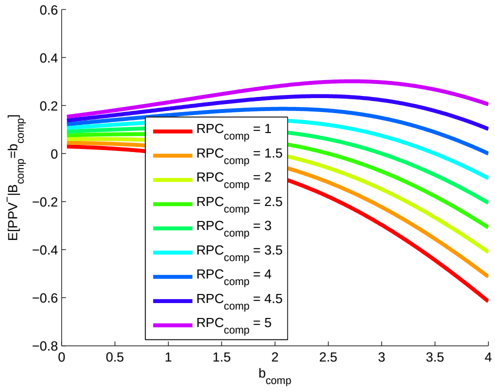
$$\frac{\partial \mathbb{E}[PPV_{comp}^{-}|B_{comp}=b_{comp}]}{\partial b_{comp}} =$$
$$\sum_{i=1}^{a+1}\left[\frac{\frac{1}{i}}{\sum_{n=1}^{a+1}\frac{1}{n}}\right]\binom{a}{i-1}\left[-(i-1)e^{-\frac{b_{comp}q_{comp}}{\lambda}}\frac{q_{comp}}{\lambda}\left(e^{-\frac{b_{comp}q_{comp}}{\lambda}}\right)^{i-2}\right.$$
$$\cdot\left(1-e^{-\frac{b_{comp}q_{comp}}{\lambda}}\right)^{a-i+1}(RPC_{comp}-b_{comp})+(a-i+1)e^{-\frac{b_{comp}q_{comp}}{\lambda}}\frac{q_{comp}}{\lambda}$$
$$\cdot\left(e^{-\frac{b_{comp}q_{comp}}{\lambda}}\right)^{i-1}\left(1-e^{-\frac{b_{comp}q_{comp}}{\lambda}}\right)^{a-i}(RPC_{comp}-b_{comp})-\left(e^{-\frac{b_{comp}q_{comp}}{\lambda}}\right)^{i-1}$$
$$\left.\cdot\left(1-e^{-\frac{b_{comp}q_{comp}}{\lambda}}\right)^{a-i+1}\right]$$

Since the roots of this formula cannot be found analytically, formula (3.7) is implemented numerically. The parameters are varied to determine the influence of each parameter. The standard parameters are as follows:

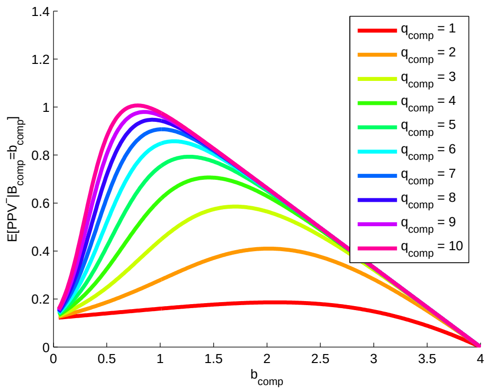
$$\begin{aligned} a &= 10, \\ RPC_{comp} &= 4, \\ q_{comp} &= 1, \\ \lambda &= 1.75. \end{aligned}$$



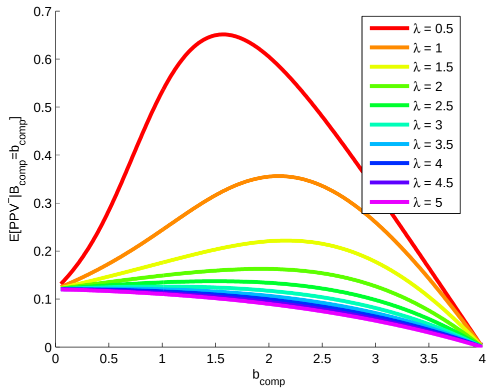
(a) Number of other advertisers a



(b) Revenue per click RPC_{comp}



(c) Quality q_{comp}



(d) Mean of exponential distribution λ

Figure 10: Variation of the parameters for exponentially distributed AdRanks

In figures (10a), (10c) and (10d) RPC_{comp} is equal to 4. Remember that CPC_{comp} is chosen to be equal to b_{comp}. That is why all lines in these figures intersect at bid 4, where the expected profit per view equals zero.
Similar as for the uniform distribution, figure (10a) shows that with a lower number of advertisers, the company can make a lower bid.
From figure (10b) it can be concluded that a higher RPC_{comp} results in a higher expected profit per view and the optimal bid becomes higher.

As for the uniform distribution, figure (10c) shows that a higher quality results in a higher expected profit per view and a lower optimal bid.
If the mean λ of the AdRanks of the other advertisers becomes higher in figure (10d), the company has to make a higher bid to reach the same position.

Recall that it still has to be discussed whether the conclusion of the position follows the conclusion of the bid. With the standard parameters the optimal position is the first position, so to bid as high as possible. With the same parameters the optimal bid is between 2 and 2.5. The difference between these results may occur because for the optimal bid $\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}]$ is used.

5.3 Normal Distribution

Next, the assumption is made that the AdRanks of the a advertisers are normally $\mathcal{N}(\mu, \sigma^2)$ distributed. For the normal distribution only the optimal bid is determined and the optimal position is skipped, since the order statistics are not known analytically.

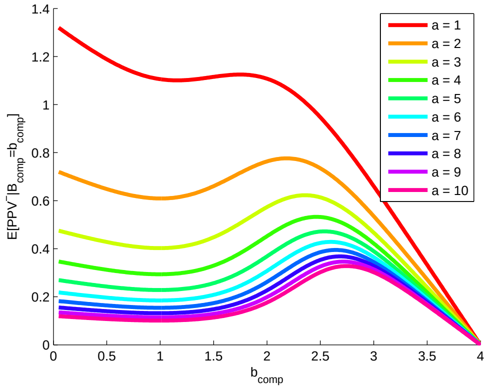
Optimal Bid

Because the optimal bid is difficult to determine analytically, this is done numerically. To do so, $p(B_{comp})$ is taken as:

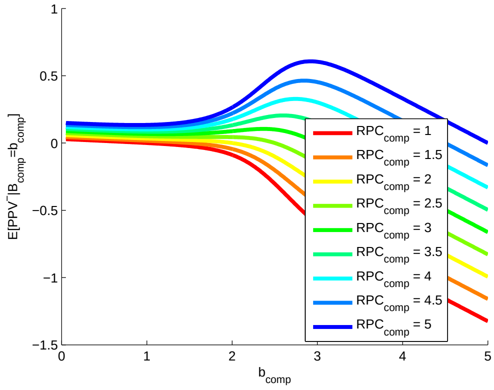
$$p(B_{comp}) = 1 - \Phi\left(\frac{q_{comp}B_{comp} - \mu}{\sigma}\right).$$

This is substituted in formula (3.7). As done for the uniform and exponential distribution, the outcomes for different values of the parameters are compared to each other. The standard parameters are chosen as follows:

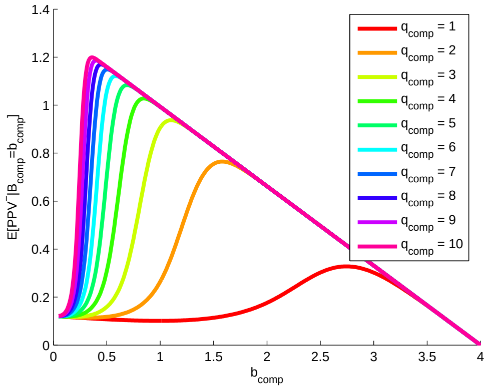
$$\begin{aligned} a &= 10, \\ RPC_{comp} &= 4, \\ q_{comp} &= 1, \\ \mu &= 1.75, \\ \sigma &= 0.6. \end{aligned}$$



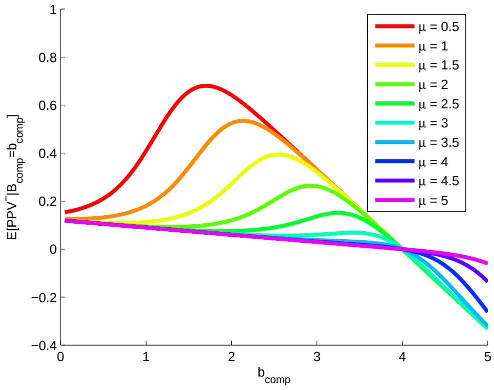
(a) Number of other advertisers a



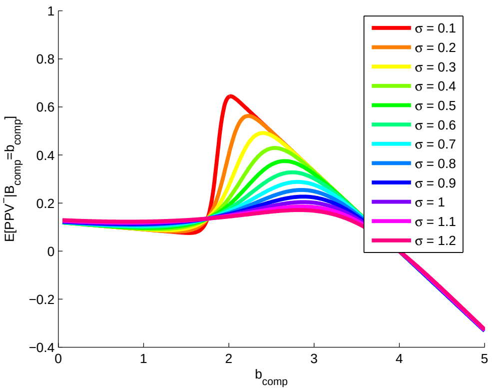
(b) Revenue per click RPC_{comp}



(c) Quality q_{comp}



(d) Mean of normal distribution μ



(e) Standard deviation of normal distribution σ

Figure 11: Variation of the parameters for normally distributed AdRanks

Just as with the exponential distribution, RPC_{comp} is equal to 4 in figures (11a), (11c), (11d) and (11e). Remember that CPC_{comp} is chosen to be equal to b_{comp}. That is why all curves in these figures have a root at bid 4.

Since the AdRanks of the advertisers are normally distributed in this case, only 2.5% of the other advertisers have an AdRank higher than $\mu + 2\sigma = 2.95$ in expectation. Figure (11a) shows that when a increases, the optimal bid approaches approximately 2.95. This makes sense because the probability that an arbitrary advertiser has an AdRank higher than this value is only 2.5%. The optimal bid increases when more advertisers are participating, because it becomes more likely that an advertiser has a high AdRank. There are only a few advertisers with a low AdRank, so it is not worthy to bid below approximately 1. The curves are decreasing in this part of the graph, since CPC_{comp} is equal to b_{comp}. Because RPC_{comp} stays the same, the cost increases with the same revenue, resulting in a lower profit per view.

The first prominent aspect in figure (11b) is that a higher RPC_{comp} gives a higher expected profit per view. Furthermore, the optimal bid for high revenues lies at approximately 2.95 like in the figure before. The first part of the graph is slightly decreasing since for low positions the click through rate is approximately the same, so it does not make a big difference in expected profit per view to bid a little higher. If the RPC_{comp} is too low, the graph is monotone decreasing, so the company should bid the minimum bid in this case.

In figure (11c) the optimal bid is decreasing for a higher quality, because a lower bid results in the same AdRank. After a certain optimal bid, the curves are decreasing since bidding higher results in a higher cost per click and the same revenue.

By varying μ, figure (11d) was created. With a lower μ the company can bid less to get the same slot, because the other advertisers also have a lower AdRank on average. This is why the optimal bid decreases for lower μ. When μ gets higher than RPC_{comp}, the curves are monotone decreasing because bidding higher than RPC_{comp} does not give a positive profit and bidding lower than μ is also not worthy for the same reasons as before.

When σ increases in figure (11e), the optimal bid increases. This could be explained because when σ increases the probability that other advertisers have a high AdRank also increases. The maximum expected profit per view also decreases when σ increases, since the company has to bid more in order to get the same revenue. The curves belonging to low σ are decreasing in the beginning because there are hardly any advertisers in this interval. Bidding more results only in a higher CPC_{comp}, while the probability of getting the last position stays really high.

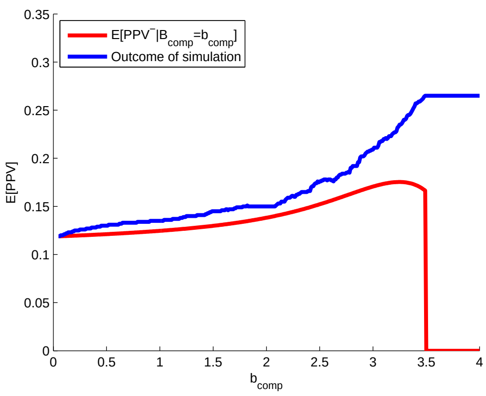
6 Simulation

In C++ a simulation program is written, that simulates auctions and determines the expected profit per view by the Monte Carlo method. The code of this program can be found in appendix A. In this simulation program AdRanks of the other advertisers are uniformly, exponentially or normally distributed. These distributions are input parameters of the program. The number of other advertisers, which is equal to the number of slots minus one, is also an input value, as are the click-through-probabilities, the revenue per click of the company and the quality of the advertisement.
For several input parameters this simulation is compared to formula (3.7). This formula was used as an estimation for the profit per view for given input parameters. For each distribution, figures are shown for several standard parameters and for a few variations of the parameters. The standard parameters are as follows:

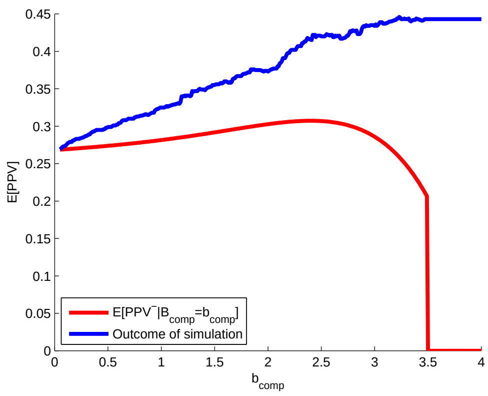
$$\begin{aligned} a &= 10, \\ RPC_{comp} &= 4, \\ q_{comp} &= 1. \end{aligned}$$

6.1 Uniform Distribution

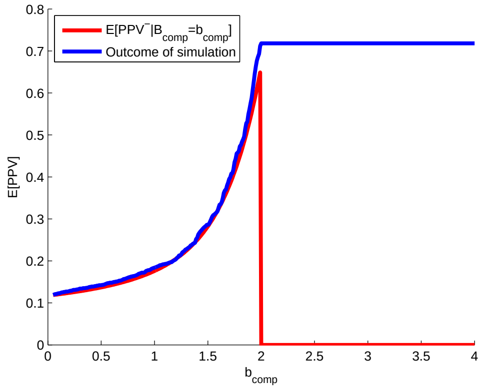
The AdRanks of the other advertisers are Uniform$[\alpha, \beta]$ distributed, with α is 0.05 and β is 3.5.



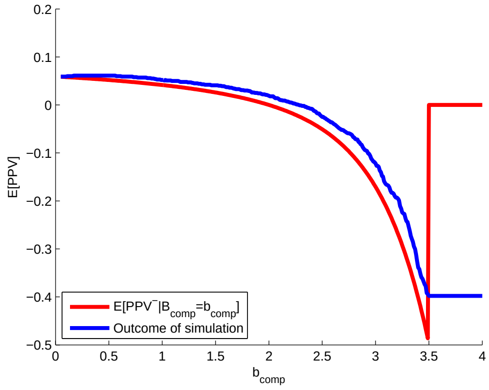
(a) Standard parameters



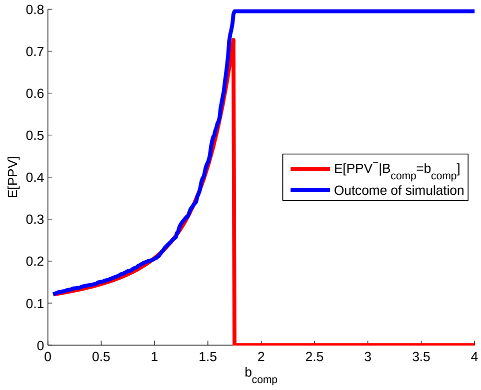
(b) Number of other advertisers $a = 5$



(c) Right boundary of uniform distribution $\beta = 2$



(d) Revenue per click $RPC_{comp} = 2$



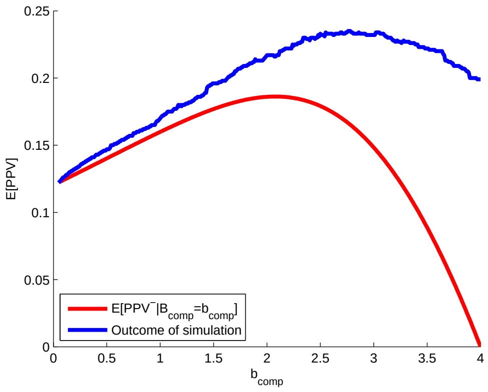
(e) Quality $q_{comp} = 2$

Figure 12: Comparison of simulation to $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}]$ with AdRanks uniformly distributed

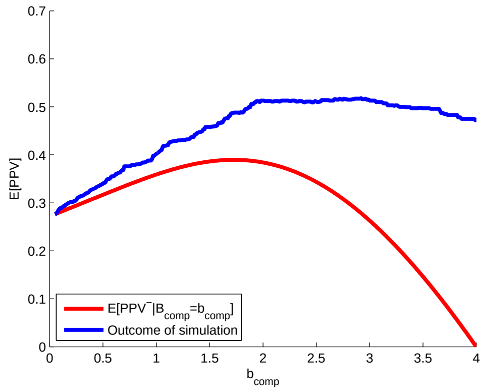
These figures show that $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}] = 0$ if $b_{comp} \geq \frac{\beta}{q_{comp}}$, since $p(b_{comp}) = 0$ in this case. As the figures show, $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}]$ is a pretty good estimate of the expected profit per view except for figure (12b). By the definition of $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}]$ it is always lower than the expected profit per view. For higher bids the approximation gets less accurate. The reason for this is that in the simulation, when b_{comp} is increasing, the company pays the AdRank of the company beneath divided by their own quality, which is bounded. On the contrary, the cost b_{comp} used in $\mathbb{E}[PPV^-_{comp}|B_{comp} = b_{comp}]$ keeps increasing.

6.2 Exponential Distribution

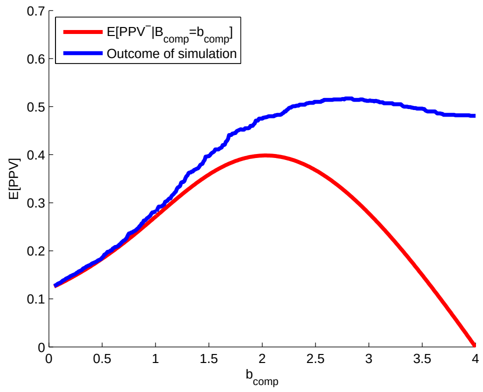
Next, the AdRanks of the other advertisers are exponentially distributed with mean $\lambda = 1.75$.



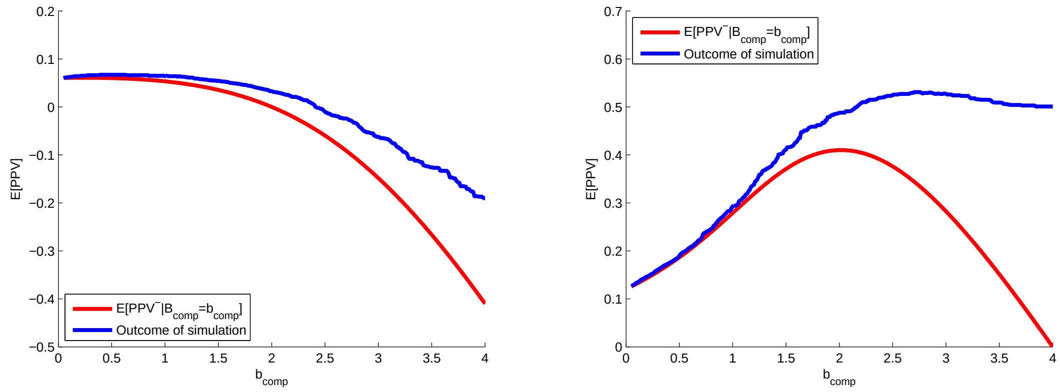
(a) Standard parameters



(b) Number of other advertisers $a = 5$



(c) Mean of exponential distribution $\lambda = 0.9$



(d) Revenue per click $RPC_{comp} = 2$

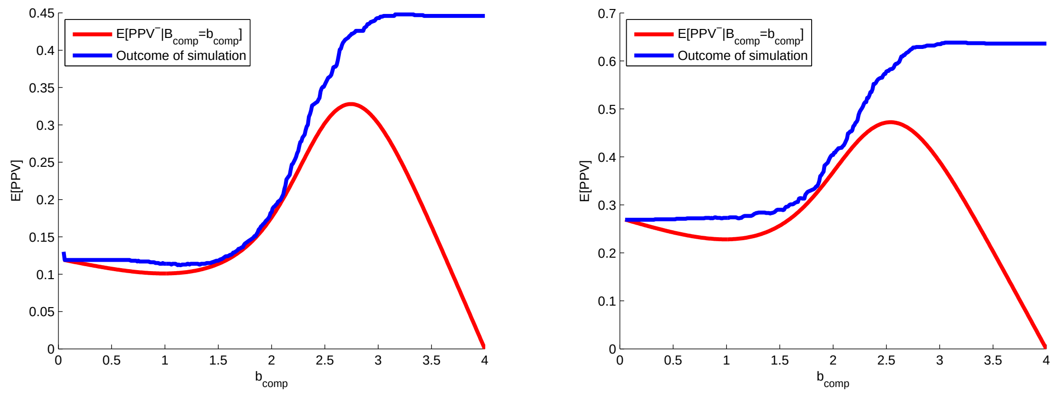
(e) Quality $q_{comp} = 2$

Figure 13: Comparison of simulation to $\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}]$ with AdRanks exponentially distributed

For the exponential distribution the figures show that for small values of b_{comp}, $\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}]$ is an accurate lower boundary for the expected profit per view. For the same reason as for the uniform distribution, the approximation gets less accurate when b_{comp} increases.

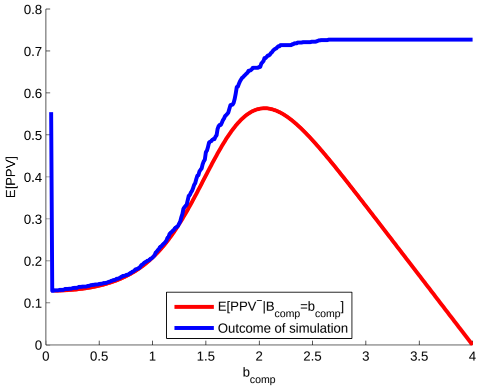
6.3 Normal Distribution

Next, the AdRanks of the other advertisers are $N(\mu, \sigma^2)$, with $\mu = 1.75$ and $\sigma = 0.6$.

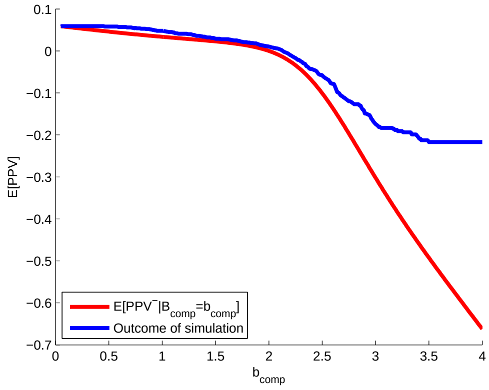


(a) Standard parameters

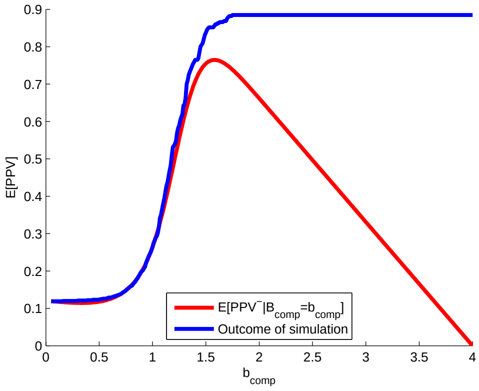
(b) Number of other advertisers $a = 5$



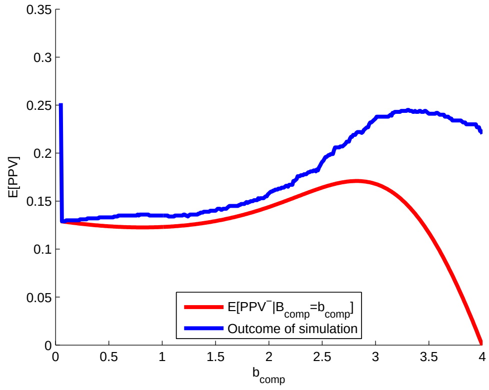
(c) Mean of normal distribution $\mu = 0.9$



(d) Revenue per click $RPC_{comp} = 2$



(e) Quality $q_{comp} = 2$



(f) Standard deviation of normal distribution $\sigma = 1.2$

Figure 14: Comparison of simulation to $\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}]$ with AdRanks normally distributed

In figure (14a), (14c) and (14f) there is a spike in the graph of the simulated PPV_{comp} at $b_{comp} = 0.05$. The reason for this is that normal random variates lower than 0.05 are adjusted to 0.05 by the simulation program. If μ decreases or σ increases the probability that an AdRank is smaller than 0.05 increases which results in a bigger spike at $b_{comp} = 0.05$. As was the case for the uniform and exponential distribution, $\mathbb{E}[PPV^{-}_{comp}|B_{comp} = b_{comp}]$ gives a good approximation for the expected profit per view for low values of b_{comp}.

7 An Algorithm to Determine the AdRanks

The profit of an advertiser depends greatly on the position of the other advertisers. In this section will be discussed how the AdRanks of other advertisers can be emperically discovered.
It is possible to calculate the AdRanks of the other advertisers by an algorithm, after making a few assumptions. Only the AdRanks of the advertisers who received a slot can be calculated with this algorithm. This algorithm terminates in s steps or less, where s is the number of slots. At each step an auction is held. The reason why the algorithm may terminate in less than s steps will be explained later.

The main assumption is that the vector of AdRanks of the other advertisers $\mathbf{r}$ is constant in time, so they do not change in the s steps. Furthermore, the company knows their quality q_{comp} and can vary their bid b_k in step k ($k = 1, 2, .., s$). So the company knows their AdRank in each step, which is $q_{comp}b_k$. The bid in each step k results in a cost per click $CPC_{comp}(b_k)$ for the company.

7.1 The algorithm

In the first step, or initialization step, choose the bid b_1 very high, such that the AdRank $b_1 q_{comp}$ is higher than the AdRanks of all other advertisers. Mathematically this means that

$$b_1 q_{comp} > r^*(1),$$

where $\mathbf{r}^*$ is the sorted descending vector $\mathbf{r}$. The resulting cost per click after the first bid, $CPC_{comp}(b_1)$, is known. Because of formula (3.1) $CPC_{comp}(b_1)q_{comp}$ equals the highest AdRank among all other advertisers, so

$$CPC_{comp}(b_1)q_{comp} = r^*(1).$$

The next step is to calculate $r^*(2)$ with information obtained by choosing a new bid b_2 smartly. The idea is to choose the bid b_2 such that the resulting AdRank $b_2 q_{comp}$ is lower than $r^*(1)$ and higher than $r^*(2)$, because then $r^*(2)$ can be calculated via the CPC formula and the obtained $CPC_{comp}(b_2)$. Because you do not know the value of $r^*(2)$ you want to bid as high as possible, but still lower than $r^*(1)$, this means that you have to bid

$$b_2 = (CPC_{comp}(b_1) - 0.01).$$

Bidding b_2 does not guarantee that the calculated AdRank is $r^*(2)$, it could be $r^*(j)$, $j > 2$. But this is only the case if $r^*(1) - r^*(2) < 0.01q_{comp}$. Then the AdRanks $r^*(1)$,..,$r^*(j)$ are very close to each other and they are in the interval $[r^*(1), r^*(1) - 0.01q_{comp})$.

The next steps are similar to the second step. So in general for step $k = 2, .., s$ the bid is:

$$b_k = (CPC_{comp}(b_{k-1}) - 0.01).$$

This results in $CPC_{comp}(b_k)$ and the next calculation is:

$$CPC_{comp}(b_{k-1})q_{comp} \quad = \quad r^*(k) \text{ if } r^*(j-1) - r^*(j) > 0.01q_{comp} \; \forall \; j = 2,..,k.$$

If the difference between two AdRanks is smaller than $0.01q_{comp}$, the algorithm skips one or more advertisers who have AdRanks in this interval. Then the algorithm will be finished in fewer steps than s and you know that some AdRanks are close to each other. Thus in this case you get a good impression of the distribution of the AdRanks.

7.2 Optimal Bidding

When the company knows the AdRanks of the other advertisers it is possible to determine an interval for the optimal bid. In this case optimal means that the maximal profit per view is gained by bidding in this interval. In this algorithm the profit per view $PPV_{comp}(i) = (RPC_{comp}(i) - CPC_{comp}(i))CTR(i)$ is dependent of the position $i = 1,..,s$. So in this case RPC_{comp} also depends on i and is not a constant like before.
When all these values from the previous formula are known by the company, $\max_i PPV_{comp}(i)$, the maximum profit per view, can be calculated. So $\arg\max_i PPV_{comp}(i)$ is the position with the highest profit. This position is acquired by choosing the bid b_{opt} optimal. To acquire the first position the company chooses the bid $b_{opt} > CPC(1) = r^*(1)/q_{comp}$. For the other positions $i > 1$ choose the bid $r^*(i)/q_{comp} < b_{opt} < r^*(i-1)/q_{comp}$.

7.3 Practicability

Outcomes, like the resulting CPC and position, of a single auction are not known in Google Adwords. The average is taken over the outcomes of all auctions in an hour. The resulting averages such as average position and average cost per click are not very precise, since all the variables are averaged and the conditions per auction could differ. This is probably not a problem for the first step but it probably is for the next steps of the algorithm. One of the reasons is that the number of advertisers in an auction may vary because of budget constraints, which leads to a (positive) variance in the average cost per click. Then it is not possible anymore to determine the other AdRanks precisely. Because of the former statement constructing a confidence interval is not possible too, because of not knowing the outcomes per auction.

7.4 Vindictive Bidding

A possible application of the algorithm is the vindictive bidding strategy. Vindictive bidding is a bidding strategy used in GSP auctions, according to [3]. In general, in this strategy the bids are known by the company. The goal of vindictive bidding is to let the competitor pay as much as possible without paying extra costs himself. By letting the competitor pay more, the company makes sure that the competitor runs out of cash more quickly. If his whole budget is spent, there is one competitor less in the auction.

With vindictive bidding a participant bids as close to the competitor above him as possible, but with the restriction that he stays under the competitor's bid. This way he will make sure that the competitor pays as much as possible for the item.
But what if other competitors also use this strategy? Then the competitor bids just below the participant, since his goal is now to maximize the participants price. He reacts by bidding just below the competitor's new bid and then the competitor reacts, etcetera. This continues until an equilibrium occurs.
Vindictive bidding might be a useful strategy when a participant knows the bids of his competitors. Unfortunately, most auctions use sealed bids to prevent this kind of strategy.

The 'bids' in a Google auction are the AdRanks of the participants. Because it is difficult in practice to determine the AdRanks of the other bidders, vindictive bidding is not a very useful strategy. An advertiser may end up having his AdRank a little bit higher than his competition, by a bad estimate. Thereby his competitors' costs are reduced and his own are raised. This is the opposite of what he wanted to achieve.

8 Conclusion

In this report a model has been presented to determine the optimal bid and position in Google AdWords, resulting in a maximum profit per view. For the AdRanks of the advertisers distributions were chosen to substitute in the model. The uniform, exponential and normal distribution were selected in this report to examine the different outcomes of these distributions.

To calculate the optimal bid a lower boundary of the expected profit per view has been derived by assuming the cost per click equal to the bid of the company. This lower boundary of the profit per view was compared to a simulation of the auction. It turned out that this lower boundary is a good estimate of the profit per view for low values of the bid. However, when the bid increases, the cost per click increases at the same rate. Since normally an advertiser pays the AdRank of the advertisement beneath divided by his own quality, the profit per view decreases when assumed that the cost per click equals the bid. In the simulation when the advertiser is on the first position, he still pays the AdRank of the advertiser below him divided by his own quality when his bid is increasing, so the profit stays constant. Still, the lower boundary of the profit per view and the outcome of the simulation show the same behaviour when several parameters are varied.

When the AdRanks of the advertisers are distributed uniformly, it turns out that the optimal position is the highest slot and the optimal bid is the highest bid, but only when the revenue per click is high enough. If the revenue per click is below the cost per click, it is best to bid the mimimum.

For an exponential distribution of the AdRanks, conclusions are more difficult to draw. A lower and upper boundary have been derived for the profit per view conditioned on the position. For the upper boundary the optimal position was calculated in an analytical way, but for the lower boundary this was not possible. This is why the optimal position, as well as the optimal bid, is determined numerically. It turned out that with the same parameters the optimal bid is not equal to the optimal position. The explanation for this could be that the profit per view for the optimal bid is a lower boundary of the exact profit per view. The optimal bid and position are dependent of several parameters, but for fixed parameters, optima can be found.

In case the AdRanks are normally distributed, only the optimal bid was examined. Again, this had to be done numerically, resulting in different outcomes for different parameters.

To fulfill the model resulting in the conclusions above, some data had to be analysed. Examination of the click through rate versus the average position, showed that the click through rate follows the Zipf's law. This is an important aspect implemented in the model.

The AdRanks of the other competitors could be calculated by an algorithm, at least in theory. If this would be possible in practice, vindictive bidding might be a useful bidding strategy.

9 Discussion

In this report several assumptions were made, some more realistic then others. The first important assumption was that the AdRanks of the advertisers are never equal. Of course in Google Adwords it can happen that two advertisers have the same AdRank. In this case Google determines which advertiser gets a higher position in a way unknown to the public. This gives the same outcome as in the case where one advertiser has an AdRank slightly higher than the other advertiser, which is done in the model. So although this assumption is not realistic, it does not have a significant effect on the model.

The second assumption made at the start of the model, is that the number of slots never exceeds the number of advertisers. This is realistic, because the number of slots depends on the number of advertisers. Google determines for each auction how many slots are shown in Google Search and this is never more then the number of advertisers. Later on in the model, the extra assumption was made that the number of slots equals the number of advertisers. In a real auction there are more participating advertisers than available slots. If there are not enough participating advertisers, Google changes the number of slots, like explained before. This assumption was made to simplify the model, although this might change the outcome, because if not all advertisers can get a slot they have to place a higher bid to get one.

In the model three different distributions for the AdRanks were examined. These three were chosen because these are common distributions. The uniform distribution was chosen to start with because it is relatively easy to analyse and it gives a good impression of the behaviour of the bid and position. The exponential and normal distributions were chosen because they are a more truthful representation of the true behaviour of the AdRanks. The normal distribution is the most realistic one of the three, because in most auctions the same type of companies participate and they want to pay approximately the same price for a slot. Thus there are not many advertisers bidding much more or much less than the average bid. Of course also the assumption of the normal distribution does not exactly hold for the AdRanks in every auction in Google Search, but it is a good approximation to base the model on. Unfortunately the optimal position with a normal distribution is not covered in this report, because to do so, the order statistics of the normal distribution are needed. This is difficult to implement. There are some approximations for this function, but the approximations found were not good enough to say something about the optimal position.

In section 3 the assumption is made that the cost per click is equal to the bid of the company. This is not very realistic, but it gives a good lower boundary for the profit per view, because the cost per click only be lower than the bid of the company. It becomes more realistic when there are more participants in the auction, since then the bids are closer to each other.

In the model the revenue per click does not depend on the position. This is probably a good approximation of the reality. In reality the revenue per click does depend on the position, since the behaviour of the user of Google Search changes during the search. For example, the user clicks on the first advertisement to look for information. After clicking on the third advertisement the user knows enough of the product he is looking for and then decides to buy the product at that website. The advertiser on the third position then gets revenue with only one click. This is why in the algorithm the revenue per click does depend on the position.

In the algorithm the assumption is made that the vector of AdRanks is constant in time. This is a realistic assumption since most companies probably do not change their bid often.

Of course not all aspects of bidding in Google Adwords are covered in this report. A closer look at other distributions can be taken to further examine the behaviour of the bid and position. The normal distribution can be solved analytically to see if the conclusions of the numerical part can be confirmed. Besides that, if more detailed data is available, the model could be tested more accurately. Furthermore, the conclusions drawn are based on the minimal profit per view. If a maximum profit per view can be derived, more conclusions can be drawn about the optimal position.
Also, if more data would be available, the average cost per click versus the average position could be further examined. The hypothesis that a higher cost per click will give a higher position could then be tested.

10 References

[1] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge U. Press, New York, 2010.

[2] Furcy Pin and Peter Key. Stochastic variability in sponsored search auctions: Observations and models. In *EC '11 Proceedings of the 12th ACM conference on Electronic commerce*, pages 1–25, New York, 2011. ACM. doi: 10.1145/1993574.1993586.

[3] Yunhong Zhou and Rajan Lukose. Vindictive bidding in keyword auctions. In *ICEC '07 Proceedings of the ninth international conference on Electronic commerce*, pages 141–145, New York, 2007. ACM. doi: 10.1145/1282100.1282130.

[4] Jon Feldman, S. Muthukrishnan, Martin Pal, and Cliff Stein. Budget optimization in search-based advertising auctions. In *EC '07 Proceedings of the 8th ACM conference on Electronic commerce*, pages 40–49, New York, 2007. ACM. doi: 10.1145/1250910.1250917.

[5] B. Kitts and B. Leblanc. Optimal bidding on keyword auctions. *Electronic Markets*, 14(3):186–201, 2004. doi: 10.1080/1019678042000245119.

[6] Michal Feldman, Reshef Meir, and Moshe Tennenholtz. Revenue enhancement in ad auctions. *Lecture Notes in Computer Science*, 7080:391–398, 2011. doi: 10.1007/978-3-642-25510-6-34.

[7] Gagan Aggarwal, Jon Feldman, and S. Muthukrishnan. Bidding to the top: Vcg and equilibria of position-based auctions. In *Proceedings of the Fourth Workshop on Approximation and Online Algorithms (WAOA)*, pages 15–28, 2006. doi: 10.1007/11970125-2.

[8] Dinesh Garg and Y. Narahari. An optimal mechanism for sponsored search auctions on the web and comparison with other mechanisms. *IEEE Transactions on Automation Science and Engineering*, 6(4):641–657, 2009. doi: 10.1109/TASE.2009.2021975.

[9] H. N. Nagaraja. *Order Statistics from Independent Exponential Random Variables and the Sum of the Top Order Statistics*, pages 173–185. Birkhuser Boston, 2006. doi: 10.1007/0-8176-4487-3-11.

11 Appendix

In this appendix both C++ and Matlab programs can be found, that are used for the simulation of our model.

Appendix A

The first program is in C++ and is used to simulate the different distributions of the AdRanks. This program calculates the expected profit for different values of the bid.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <iostream>
#include <fstream>
#include <ctime>
#include <cstdlib>
using namespace std;

FILE  *infile, *outfile;

float exp(float mean,float rand);
float uniform(float min, float max, float rand);
float lcgrand(int stream);
float normal(float mu, float sigma, float rand1, float rand2);
void  zipfslaw(int Aa, int Ss, float s);

#define NUM_OF_SIMULATIONS 100

#define S 11
#define A 10

float r[A+1];
float r_star[A+2];
float r_star_adv[A+2];
float cpc[A+1];
float prob[S+1];
float er[A+1];
float av_er[10000000+1];
float av_r_star[10000000+1];
float tot_av_er[10000000];
int main()
{

        float   rpc = 4.50;
        float   q_comp = 1;

        int     maxbodmaal100 = 500; //ceil(rpc*100)+90;


        for(int i=5; i<=maxbodmaal100; i++)
        {
                tot_av_er[i]=0;
        }
        for(int sim_num = 1; sim_num <= NUM_OF_SIMULATIONS; sim_num++)
        {

                for(int i=1; i<=A; i++)
                {
                        //r[i] = uniform(.05, 2.0, lcgrand(sim_num));
                        //r[i] = exp(2.0, lcgrand(sim_num));
                        r[i] = normal(2.1, .3, lcgrand(sim_num), lcgrand(sim_num));
                }
                zipfslaw(A,S,1);

                for(int m=5; m<=maxbodmaal100; m++)
                {
                        r[0] = q_comp*(m/100.0);

                        for(int j=0; j<=A; j++)
                        {
                                int counter = 0;
                                for(int i=0; i<=A; i++)
                                {
                                        if(r[j]<r[i])
                                        {
                                                counter++;
                                        }
                                }
                                counter++;
                                r_star[counter]=r[j];
                                r_star_adv[counter]=j;
                        }

                        int j=0;
                        int n;
                        for(int i=1; i<=S; i++)
                        {
                                if(r_star_adv[i] == j)
                                {
```

```
                    n=i;
                }
            }
            if(n==S)
            {
                cpc[j]=0.05;

            }
            if(n<S)
            {
                cpc[j] = r_star[n+1]/q_comp;
            }
            if(n>S)
            {
                cpc[j] = 0;
            }

            er[m] = (rpc-cpc[0])*prob[n];

            tot_av_er[m] += er[m];
        }

            int h = 5;
            for(int j=5; j<=maxbodmaal100; j++)
            {
                if(er[j]>er[h])
                {
                    h=j;
                }
            }
            float max_er    = er[h];
            float opt_bid   = h*1.0/100;

            //cout << opt_bid << max_er << endl;

    }
    outfile = fopen("bosor_simulatie2.out", "w");
    for(int m=5; m<=maxbodmaal100; m++)
    {
        fprintf(outfile, "\n m \%10.3f", m/100.0);
        //fprintf(outfile, "\n ER \%8.3f \n", av_er[m]);
        fprintf(outfile, "\n ER \%8.3f \n", tot_av_er[m]/(1.0*
              NUM_OF_SIMULATIONS));
    }

    fclose(outfile);
}

float uniform(float min, float max, float rand)
{
    float f = min+(max-min)*rand;
    return f;
}

float exp(float mean, float rand)
{
    /* exponential variate with mean mean from (0,1) random number rand, using
          inverse transformation method */
    float f = -mean*log(rand);
    return f;
}

float normal(float mu, float sigma, float rand1, float rand2)
{
    float z1 = sqrt(-2*log(rand1))*cos(2*M_PI*rand2);
    float z2 = sqrt(-2*log(rand1))*sin(2*M_PI*rand2);
    if(sigma*z1+mu<0.05)
    {
        return 0.05;
    }
    else
    {
        return sigma*z1+mu;
    }
}

void zipfslaw(int Aa, int Ss, float s)
{
    float sum = 0.0;
    for(int i=1; i<=Ss; i++)
    {
        float j=1.0*i;
        prob[i]=1.0/j;
        sum = sum + prob[i];
    }
    if(Aa+1>Ss)
    {
        for(int i=Ss+1; i<=Aa+1; i++)
        {
            prob[i]=0;
        }
    }
    for(int i=1; i<=Ss; i++)
    {
        prob[i] = prob[i] / sum;
    }
}
```

```c
//Random Generator

#define MODLUS 2147483647
#define MULT1       24112
#define MULT2       26143

static long zrng[] =
{         1,
 1973272912, 281629770,  20006270,1280689831,2096730329,1933576050,
  913566091, 246780520,1363774876, 604901985,1511192140,1259851944,
  824064364, 150493284, 242708531,  75253171,1964472944,1202299975,
  233217322,1911216000, 726370533, 403498145, 993232223,1103205531,
  762430696,1922803170,1385516923,  76271663, 413682397, 726466604,
  336157058,1432650381,1120463904, 595778810, 877722890,1046574445,
   68911991,2088367019, 748545416, 622401386,2122378830, 640690903,
 1774806513,2132545692,2079249579,  78130110, 852776735,1187867272,
 1351423507,1645973084,1997049139, 922510944,2045512870, 898585771,
  243649545,1004818771, 773686062, 403188473, 372279877,1901633463,
  498067494,2087759558, 493157915, 597104727,1530940798,1814496276,
  536444882,1663153658, 855503735,  67784357,1432404475, 619691088,
  119025595, 880802310, 176192644,1116780070, 277854671,1366580350,
 1142483975,2026948561,1053920743, 786262391,1792203830,1494667770,
 1923011392,1433700034,1244184613,1147297105, 539712780,1545929719,
  190641742,1645390429, 264907697, 620389253,1502074852, 927711160,
  364849192,2049576050, 638580085, 547070247 };

float lcgrand(int stream)
{
    long zi, lowprd, hi31;

    zi     = zrng[stream];
    lowprd = (zi & 65535) * MULT1;
    hi31   = (zi >> 16) * MULT1 + (lowprd >> 16);
    zi     = ((lowprd & 65535) - MODLUS) +
             ((hi31 & 32767) << 16) + (hi31 >> 15);
    if (zi < 0) zi += MODLUS;
    lowprd = (zi & 65535) * MULT2;
    hi31   = (zi >> 16) * MULT2 + (lowprd >> 16);
    zi     = ((lowprd & 65535) - MODLUS) +
             ((hi31 & 32767) << 16) + (hi31 >> 15);
    if (zi < 0) zi += MODLUS;
    zrng[stream] = zi;
    return (zi >> 7 | 1) / 16777216.0;
}


void lcgrandst (long zset, int stream)
{
    zrng[stream] = zset;
}
```

Appendix B

For the exponential distribution, a graph of the optimal position is made with the following program in Matlab. The expected profit per view is shown together with the lower and upper boundary.

```matlab
close all
clear all
a=10; B=4; A=0.05; RPC = 4; r=0; q=1; L=1.75;
PPV1=0; PPV2=0; PPV3=0;
x=[]; v=[]; w=[]; z=[]; y=[];

for n=1:(a+1)
    r = r+(1/n);
end

for i=1:a+1
    f=0;
        PPV1 = (RPC-(log(a+1)-log(i))/(L*q))*((1/i)/r);
        PPV2= (RPC-(log(a)-log(i-1))/(L*q))*(1/i)/r;
        if (i==a+1)
            f=A;
        else
            for k=1:a-i+1
            f=f+1/(a-k+1);
            end
        end
        PPV3 = (RPC-f/(L*q))*((1/i)/r);
v = [v i]; w = [w PPV1]; z = [z PPV2]; y = [y PPV3];
end
hold on
plot(v,w,'r','Linewidth',3);
plot(v,z,'b','Linewidth',3);
plot(v,y,'g','Linewidth',3);
xlabel('i');
ylabel('E[PPV|Y=i]');
legend('Upper boundary','Lower boundary','Original','Location', 'Best')
```

Appendix C

Also for the optimal bid a program in Matlab is made. This program is shown below for a uniform distribution of the AdRanks. If the AdRanks are distributed otherwise, only the probabilities change. The RPC_{comp} is varying in this case, of course another parameter can be varied by making a few changes to the program.

```
close all
clear all
a=10; C=5; B=3.5; A=0.05; q=1;
cc=hsv(10);

for RPC=1:0.5:5
r=0; v=[]; w=[];
    for i=1:(a+1)
    r = r+(1/i);
    end

    for b=A:0.01:C
        y=0;
        for i=1:(a+1)
            x = nchoosek(a,i-1);
            if B>b*q
            y = x*(((B-b*q)/(B-A))^(i-1))*((1-(B-b*q)/(B-A))^(a-i+1))*(RPC - b)*((1/
                i)/r)+y;
            else
            y=y;
            end
        end
    v = [v b];
    w = [w y];
    end

hold on
k=RPC*2-1;
plot(v,w,'Color', cc(k,:),'Linewidth',3)
xlabel('b_{comp}');
ylabel('E[PPV^-|B_{comp}=b_{comp}]');
legendInfo{k} = ['RPC_{comp} = ' num2str(RPC,2)];
end
legend(legendInfo, 'Location', 'Best');
```