

# CHALMERS



Optimizing on-line display-ad allocation subject  
to advertiser budget constraints

*Master of Science Thesis in the Programme Computer Science:  
Algorithms, Languages and Logic*

ANDREAS GRANSTRÖM

Chalmers University of Technology  
University of Gothenburg  
Department of Computer Science and Engineering  
Göteborg, Sweden, May 2012

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Optimizing on-line display-ad allocation subject to advertiser budget constraints

ANDREAS GRANSTRÖM

©ANDREAS GRANSTRÖM, May 2012.

Examiner: DEVDATT DUBHASHI

CHALMERS UNIVERSITY OF TECHNOLOGY  
UNIVERSITY OF GOTHENBURG  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Cover: In the foreground, a simplified objective function for maximizing publisher revenue. In the background, predicted (solid line) and observed (circles) traffic for a placement.

Department of Computer Science and Engineering  
Göteborg, Sweden February 2012

---

## Abstract

It is common among on-line publishers to monetize their visitors by displaying advertisements. To do so, they have the option to use systems called *display-ad exchanges* to help decide which advertisements are shown to each visitor.

The key challenge is to allocate advertisements to viewers in a real time setting. This thesis develops a model that optimizes how the display-ad exchange spends the budget of advertisers in order to maximize the revenue of the publisher. This problem is virtually unaddressed in literature.

The model is constructed by combining an off-line linear programming model with a linear regression model for web traffic prediction. This combination renders a solution from which it is possible to measure *return-on-investment* values that can be used by the display-ad exchange to increase the publisher revenue.

The thesis develops a greedy Baseline algorithm that simulates key characteristics of a real display-ad exchange. Comparing the return-on-investment heuristic with the Baseline for a set of real data, shows a 2% increase in publisher revenue. This increase is achieved by spending advertiser budgets more efficiently. The off-line linear programming model shows theoretical revenue improvements in the region of 6%, and that this figure depends on how many advertisers completely consume their budgets.

---

## **Acknowledgements**

I would like to thank my two advisors, David Burke at Admeta and Chiranjib Bhattacharyya at Chalmers. They have both provided support and encouragement through the work on this thesis.

I would also like to thank the company Admeta. They welcomed me to their office and provided the data and information necessary for me to complete this thesis.

## Contents

<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem definition . . . . .	2
1.2 A budget optimization model . . . . .	3
<b>2 Background</b>	<b>5</b>
2.1 Domain description . . . . .	5
2.2 Motivational example . . . . .	7
2.3 Problem definition . . . . .	7
2.4 Related work . . . . .	9
2.5 Summary . . . . .	9
<b>3 Data model</b>	<b>10</b>
3.1 Data model format . . . . .	10
3.2 Output data format . . . . .	12
3.3 Case-study data . . . . .	13
3.4 Summary . . . . .	14
<b>4 A formal Baseline algorithm</b>	<b>15</b>
4.1 An ad exchange algorithm . . . . .	15
4.2 The Baseline algorithm . . . . .	16
4.3 Comparison to production . . . . .	17
4.4 Baseline budget distribution and performance . . . . .	17
4.5 Summary . . . . .	18
<b>5 Off-line budget optimization</b>	<b>19</b>
5.1 Linear programming . . . . .	19
5.2 Model input . . . . .	19
5.3 Objective . . . . .	20
5.4 Constraints . . . . .	20
5.5 Baseline comparison . . . . .	21
5.6 Budget limit impacts . . . . .	23
5.7 Additional model constraints . . . . .	23
5.8 Summary . . . . .	23

<b>6</b>	<b>Traffic prediction</b>	<b>24</b>
6.1	Problem definition . . . . .	24
6.2	Input data . . . . .	25
6.3	Performance measurements . . . . .	25
6.4	Weekly trends . . . . .	26
6.5	Hour model . . . . .	28
6.6	Day prediction model . . . . .	30
6.7	Results . . . . .	31
6.8	Results on case-study data . . . . .	31
6.9	Summary . . . . .	33
<b>7</b>	<b>On-line budget optimization</b>	<b>34</b>
7.1	Problem definition . . . . .	34
7.2	Order classification . . . . .	35
7.3	Return on investment . . . . .	35
7.4	The return-on-investment heuristic . . . . .	36
7.5	Results . . . . .	36
<b>8</b>	<b>Conclusion</b>	<b>41</b>
8.1	Further work . . . . .	42
<b>Bibliography</b>		<b>43</b>
<b>Glossary</b>		<b>44</b>

## List of Figures

1.1	Example of search-based advertisements . . . . .	1
1.2	Example of on-line advertisements . . . . .	2
1.3	The budget optimization model. . . . .	4
2.1	Two placements at the website of a publisher. . . . .	5
2.2	Example of creative material . . . . .	6
2.3	Example of text material . . . . .	6
2.4	The budget optimization model within an ad exchange . . . . .	8
3.1	The ad exchange data model. . . . .	10
3.2	eCPM over time for an <i>order</i> $\times$ <i>placement</i> combination. . . . .	12
3.3	eCPM over different placements for an <i>order</i> $\times$ <i>time</i> combination. . . . .	12
4.1	Baseline budget utilization example . . . . .	17
5.1	LP budget utilization example #1 . . . . .	22
5.2	LP budget utilization example #2 . . . . .	22
6.1	Historic impressions for a placement belonging to MID . . . . .	24
6.2	Impressions for popular placements . . . . .	26
6.3	Weekly trend analysis . . . . .	27
6.4	Hourly trend analysis . . . . .	28
6.5	Weekly prediction residue. . . . .	29
6.6	Daily trend analysis . . . . .	30
6.7	Example of traffic predictions . . . . .	32
6.8	Example of difficult to predict patterns . . . . .	32
7.1	The budget optimization model. . . . .	34
7.2	ROI deviation per order and placement over time . . . . .	37
7.3	Example order #1 budget spending . . . . .	39
7.4	Example order #2 budget spending . . . . .	39

## List of Tables

2.1 Greedy example of material selection . . . . .	7
3.1 Output example . . . . .	13
3.2 Impressions statistics for MID over the 13 days. . . . .	13
3.3 Order count for MID over the 13 days. . . . .	13
3.4 Placement count for MID over the 13 days. . . . .	13
3.5 Material count for MID over the 13 days. . . . .	13
3.6 Impressions statistics for LARGE over the 11 days. . . . .	14
3.7 Order count for LARGE over the 11 days. . . . .	14
3.8 Placement count for LARGE over the 11 days. . . . .	14
3.9 Material count for LARGE over the 11 days. . . . .	14
5.1 LP improvement for MID & LARGE compared to Baseline . . . . .	21
5.2 Optimization potential compared to budget consumption . . . . .	23
6.1 MPE measurements for $f_{week}$ predictions. . . . .	27
6.2 Performance of week prediction model . . . . .	28
6.3 MPE's of hour prediction model . . . . .	29
6.4 Hour prediction model benchmark . . . . .	29
6.5 Day prediction model performance . . . . .	31
6.6 Day prediction model improved with less training . . . . .	31
6.7 Predicted placement quantity . . . . .	32
6.8 Traffic prediction eval for Case Study data . . . . .	32
6.9 Case Study prediction benchmark . . . . .	33
7.1 ROI deviation per order and placement over time . . . . .	36
7.2 ROI per order and time over placements . . . . .	37
7.3 Improvement of ROI method for MID . . . . .	38
7.4 Improvement of ROI method for LARGE . . . . .	38
7.5 ROI revenue as fraction of optimum for MID . . . . .	38
7.6 ROI revenue as fraction of optimum for LARGE . . . . .	38
7.7 Impression consumption for ROI method . . . . .	38



## Introduction

On-line publishers, such as Göteborgs Posten<sup>1</sup> and Aftonbladet<sup>2</sup> can monetize their visitors by displaying advertisements. They have the option to use advanced systems called ad exchanges to help them decide which advertisements are shown to their visitors.

There exist two main types of ad-exchanges:

**Search-ad exchanges (Figure 1.1)** like Google AdWords, pick which advertisements are shown to each visitor based on keywords provided by the visitor. Most literature available focuses on search-ad exchanges.

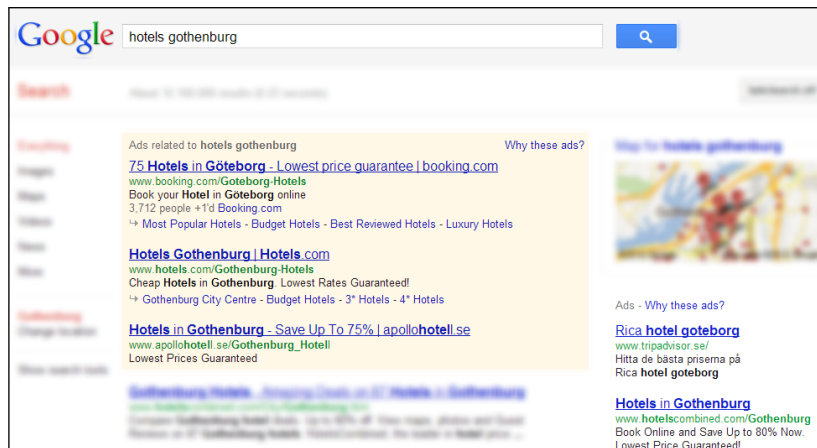


FIGURE 1.1: Example of advertisements picked by a search-based ad exchange. The user has provided two keywords “hotels” and “gothenburg”. The search-ad exchange shows advertisements relating to the two keywords, in this case it is recommendations for hotels in the Gothenburg area.

**Display-ad exchanges (Figure 1.2)** decide what advertisements are shown to each visitor based on relevant available parameters such as: characteristics of the advertisements and the advertisers; contextual information of the website; and visitor behaviour and demographics if available.

This thesis is about maximizing the revenue of publishers that use display-ad exchanges. In a display-ad exchange (hereafter referred to as *ad exchange*)

<sup>1</sup><http://gp.se>

<sup>2</sup><http://www.aftonbladet.se>

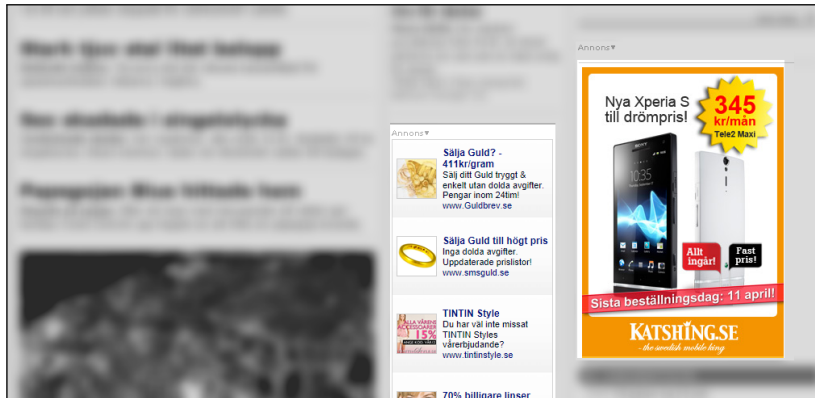


FIGURE 1.2: Advertisements at the website of an on-line publisher. Advertisements are selected subject to contextual information, characteristics of the advertisements and other parameters available to the display-ad exchange.

the publisher has a set of placements on his website; advertisers bid money to have their advertisements shown on the publishers placements; the ad exchange picks the winning bidder and the winner gets their advertisement shown. The bid is what renders revenue to the publisher.

For both types of ad exchanges there exists literature attempting to find optimal bidding strategies for the advertisers. In this thesis we diverge from the perspective of the advertiser and assume the perspective of the *publisher*, something not commonly done.

## 1.1 Problem definition

The publisher wants to maximize his revenue. The revenue comes from the advertisers who place bids to have their advertisements shown. The advertiser pays the publisher either per thousand views of an advertisement, when a visitor clicks the advertisement, or performs some other action on the advertiser's site after viewing or clicking on the advertisement. The advertisers often also have a limited budget to spend and as the ad exchange greedily picks the advertisement that has the highest expected revenue, this does not take into account the fact that it might be more profitable to save the advertiser's budget for a different visitor, placement or time when the expected revenue of this advertisement might be higher. We look at the problem of *advertiser budget optimization from the perspective of the publisher*. Optimizing how the publisher utilizes the budgets of advertisers means that we want to consume the budget of the advertisers using as few views as possible.

The idea is that some advertisers will completely consume their budgets during some time period, and some advertisers will not. By reducing the amount of web traffic used by advertisers consuming their budgets, the remaining advertisers will have more web traffic available to them. The effect is increased revenue for the publisher.

## Difficulties

The problem is difficult for a set of reasons: (i) publishers receive large amounts of web traffic each day (tens or hundreds of millions of advertisements must be displayed), and each visitor must be served in real-time; (ii) publishers have a large set of placements and advertisers, making the problem of picking advertisements for placements combinatorially difficult; (iii) visitors arrive to the publisher in an on-line fashion. The pattern in which the visitors arrive is difficult to predict. Consequently, this makes it difficult to decide if to show an advertisement now, or save it for later; (iv) ad exchanges are dynamic systems: advertisers, placements and bids can enter and leave the system at any time.

The problem is both relevant and important since even a slight increase in publisher revenue subsequently renders an increase in revenue for the ad exchange provider. It also strengthens the competitive edge for the ad exchange, making the system more marketable and appealing to on-line publishers.

## 1.2 A budget optimization model

In this thesis we will present a complete budget optimization model, something that has not been done before. We will show a theoretical increase in revenue of 6% and an on-line heuristic that increases the revenue by 2% in a real-time production-like system using real data. Furthermore, we show that the advertisers that completely consume their budget do so in 8% fewer advertisement views, indicating that we are also better at choosing the right target audience to show the advertisements to. The budget optimization model is plug-in by nature, meaning that it does not depend on the inner workings of the ad exchange to be usable.

In Chapter 2 we give a full description of the problem and its domain. We explain the input data that we have used in Chapter 3. All data used in this thesis is provided by Admeta<sup>3</sup>. Admeta is a Gothenburg based company that develops a display-ad exchange called Tango.

To get a measure of the potential for optimization, we define a greedy baseline algorithm in Chapter 4. The Baseline algorithm closely resembles the behavior of a real ad exchange.

In Chapter 5 we look at the problem from an off-line perspective, i.e., we temporarily remove the on-line part of the problem and assume that we have the observed web traffic for a given day. From this perspective the problem is strictly combinatorially difficult. We develop a *linear programming* (LP) model for solving the off-line optimization. The LP gives a revenue upper bound and indicates that there is potential for increased revenue. The output of the LP is an optimal, static *placement*  $\times$  *advertisement*  $\times$  *time* allocation for received web traffic.

We will see that the increase in revenue rendered by the off-line optimization compared to the greedy Baseline algorithm is largely dependent on how many advertisers in the ad exchange completely consume their budget.

With the LP in place we define a linear regression model in Chapter 6 for predicting website traffic. Using the predictions as input to the LP we can get an estimation of what would be an optimal allocation for the next day. We

---

<sup>3</sup><http://www.admeta.com>

will see that the linear regression model can render fairly accurate predictions for the most significant placements, but that the dynamics of the ad exchange present problems when we do large-scale predictions.

Since the output of the LP is a static allocation of web traffic, even if it is an optimal solution for the prediction, it is not something that is immediately usable in the on-line scenario. Since placements, orders and bids can be added, removed and changed at any time during the days, the heuristic must be flexible enough to handle these changes.

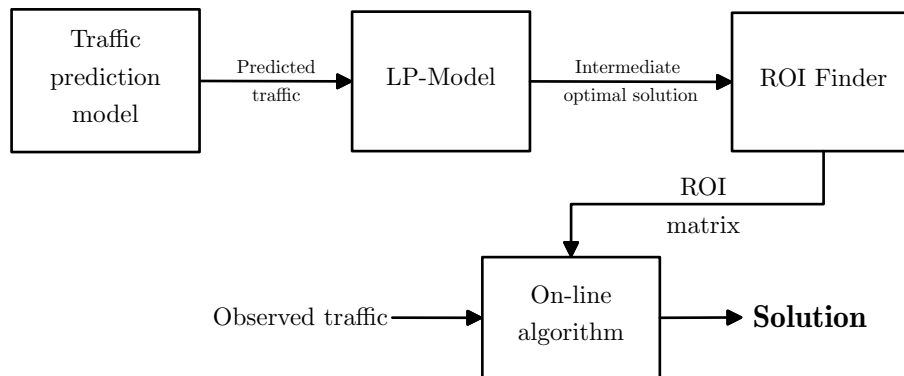


FIGURE 1.3: The budget optimization model.

Chapter 7 will present the final on-line optimization heuristic. By taking the static LP solution for the predicted web traffic, and measuring *return on investment* (ROI) values, we can use the ROI values to adjust the bids of advertisers for use in the on-line scenario. This approach is flexible enough since it does not depend on some static allocation, making it useful in practice.

The complete model can be seen in Figure 1.3. The budget optimization model reduces the amount of web traffic used by advertisers completely consuming their budgets by 8%, and increases the total revenue of the publisher by 2%.

## Background

In this chapter we will give a comprehensive description of the domain of ad exchanges; a formal problem definition and a brief description of related work.

### 2.1 Domain description

At the highest level in the ad exchange, we have publishers. Publishers own websites, and they define areas on these websites where they allow the ad exchange to place advertisements. These areas are called *placements*.

#### Placements

Placements are rectangular areas on websites where advertisements can be shown. If we hide the advertisements in Figure 1.2 we can observe the placements in Figure 2.1.

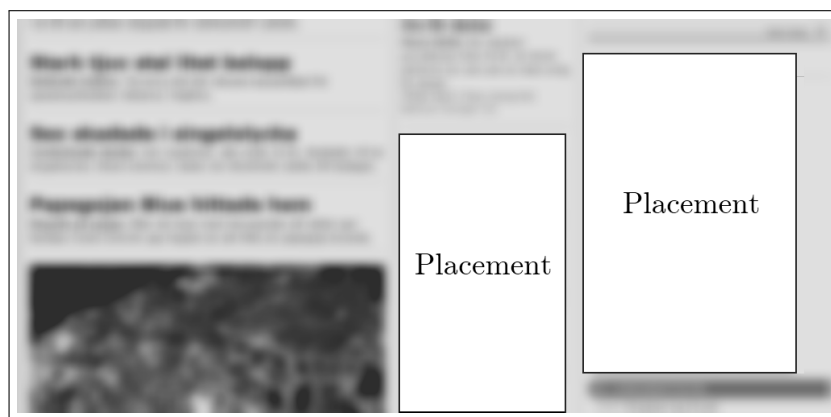


FIGURE 2.1: Two placements at the website of a publisher.

It is the job of the ad exchange to decide to which visitor and on what placement an advertisement is shown. Advertisements that can be shown on placements are called *materials*.

#### Materials

Materials are advertisements that can be allocated to placements by the ad exchange. We have two different types of materials:



FIGURE 2.2: An example of a creative material.



FIGURE 2.3: An example of a text material.

**Creative materials (Figure 2.2)** are images or something more dynamic, such as an Adobe Flash animation. Creative materials cover whole placements, i.e., we can fit at most one Creative material per placement.

**Text materials (Figure 2.3)** are a short piece of text that may be coupled with a small static image. Depending on the size of placements, it is possible to fit many text materials on one placement. Each placement has its own *text material capacity* defining how many text materials can fit at the same time. As not all placements accept text materials, the text material capacity can be zero.

## Impressions

When a visitor downloads (views) the publishers website, this is called an *impression*. In the context of this thesis we will count *placement impressions*. That is, if a website has two placements, and a single visitor enters the site, we will count two impressions, one for each of the placements. Placement impressions are independent of how many materials that are actually appearing on the placement. This allows us to abstract away from the concept of websites, and only consider a set of placements. This is useful since publishers can have multiple websites.

## Orders

Advertisers supply the publisher with materials by placing *orders*. An order is a set of materials coupled with bids (how much the advertiser is willing to pay to have their materials shown) and a budget. The ad exchange guarantees that the advertiser will never be charged more money than his budget or his bid.

## Expected Cost per Milli

When choosing which material(s) to display, the ad exchange operates in terms of *estimated Cost per Milli* (eCPM). The eCPM is the expected revenue from

showing a material one thousand times. The eCPM for each material is calculated for each impression by combining the bids of the advertisers with the likelihood that a click (or some other action that the advertiser is paying for) is performed following display of the material. For the remainder of this thesis, we will use eCPM as a direct representation of publisher revenue.

The ad exchange picks which materials to put on each placement in a *real-time* environment. This means that for each impression the system picks the highest rated eCPM material in sequence until all placements have materials assigned to them. *Because orders have limited budgets, this is not an optimal heuristic for picking materials.* We illustrate this with an example of greedily picking a material for a placement in the next section.

## 2.2 Motivational example

In this example, we have materials A and B with different budgets and *eCPM* over time, and two impressions. Note that we only consider one placement in this example.

For the first impression, material A will be selected, since A has higher *eCPM* than B. For the second impression, A still has the highest *eCPM*, but not enough budget remaining, so we are forced to choose material B, with an *eCPM* of 1.

This renders a suboptimal revenue of 3, compared to the optimum of 6.

TABLE 2.1: Example why greedy selection is suboptimal. Winning materials are highlighted in bold.

Material	Budget amount	Impression	<i>eCPM</i> (A)	<i>eCPM</i> (B)
A	5	1	<b>2</b>	1
B	10	2	5	<b>1</b>

## 2.3 Problem definition

*We want to maximize publisher revenue by optimizing how the budgets of advertisers are spent in an existing on-line algorithm.*

More precisely: we want to maximize publisher revenue within the context of an existing ad exchange. We want to find a heuristic that works for the case where impressions arrive to the publisher in an on-line fashion, and we must decide *per impression* what material is shown on what placement in order to maximize the total revenue of the publisher subject to taking careful consideration of how advertiser budgets are spent.

We have identified a set of sub-problems, or parts, that each needs to be solved.

1. A formal Baseline for measuring optimization performance.
2. Off-line budget optimization. It is important to note that the output of this part can not only be a revenue figure, but must be able to provide an optimal *placement*  $\times$  *material*  $\times$  *time* allocation.

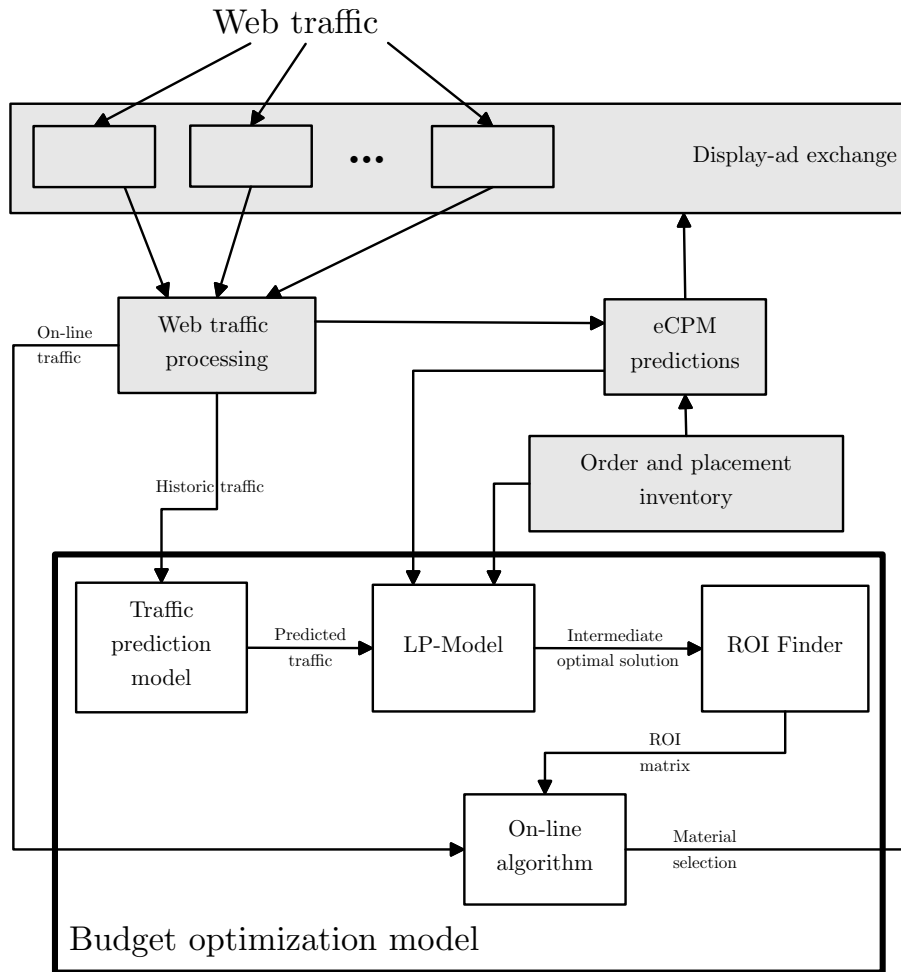


FIGURE 2.4: The budget optimization model within the context of an ad exchange. In this thesis we develop the parts in the outlined box labeled “Budget optimization model”.

3. Prediction of placement impressions (traffic prediction).
4. Development of a heuristic for the on-line budget optimization.

To solve the problem, we will build a complete budget optimization model. The parts of the model should be as interchangeable as possible. Further, the runtime of the budget optimization model should preferably be in the region of minutes - so that the publisher can run the system frequently to take advantage of the latest eCPM and traffic predictions.

Figure 2.4 shows how our model will fit into the ad exchange.

### Limitations

The scope of this thesis is quite broad. While the main objective is to maximize publisher revenue subject to advertiser budget constraints, we do have



some limitations. Primarily, we want our solution to be independent of the ad exchange system. That is, we are not attempting to re-invent an ad exchange, and we are not interested in *how* the eCPM predictions are done. Hence we will assume that the eCPM predictions are perfect and always available.

The choice of methods used for implementing each of the parts of the budget optimization model is subject to two main motivations: (1) they are fairly quick and intuitive to develop and implement; and (2) if an initial approach gives promising results, it provides a motivation for adding more sophistication later.

As stated in section 2.3, the runtime of the parts are of significance, but it is not something that we have spent a great amount of effort trying to reduce. As long as the runtime has (in our estimate) been reasonable, we have been content.

## 2.4 Related work

When researching literature on the subject of on-line advertisement and ad exchanges, one will find that there are a large amount of material available, but that a lot of the literature takes the perspective of the advertiser ([FHK<sup>+</sup>10, BCI<sup>+</sup>07, ZCL08]), trying to find an optimal strategy for bidding in a system outside of his control.

Chen et al., in a recent paper [CBAD11] attempts to optimize display-ad allocation by developing an LP model for observed (recorded) data. They present an on-line algorithm that is based on solving the dual to the LP to get values that help them adjust the eCPMs. Further, they explore methods of Control Theory to adapt the eCPM adjustment parameters while the on-line algorithm is running.

A more general method of budget optimization is presented in [MNS07] by Mahdian et al. where they assume an "oracle" that they call EST, which gives an unreliable estimate on which material to show. By adjusting parameters on how much they trust EST they claim to achieve better results than either being completely trusting or completely dismissive of EST.

## 2.5 Summary

We have presented the domain of display-ad exchanges. We have shown that the current way of selecting materials for placements renders suboptimal publisher revenue due to the ad exchange not considering how it spends the budget of advertisers.

While related work is sparse, there is some. This thesis contributes to the existing work by presenting a complete budget optimization model that is independent of the inner workings of the ad exchange.

In the next chapter we will define how we represent the data of the ad exchange and publishers.

## Data model

The previous chapter presented the budget optimization problem and its domain. In this chapter we describe the data model that will be used as input for each of the parts of the budget optimization model.

We will use real data from two different publishers called MID and LARGE.

### 3.1 Data model format

We have developed a unified data model, as seen in Figure 3.1, that can be used as input by the methods presented later in this thesis. The data model captures essential characteristics of a real ad exchange.

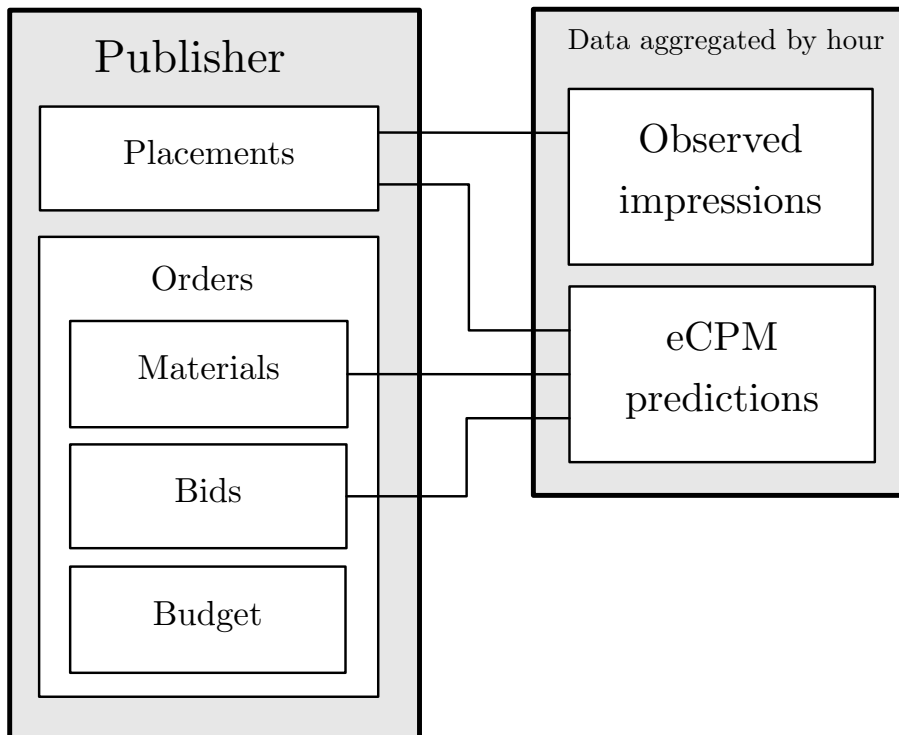


FIGURE 3.1: The ad exchange data model.

In the data model, the publisher is the user of the ad exchange. The publisher has a set of placements and a set of orders provided by advertisers.

### Orders

Each order contains a set of materials, bids and a budget.

The materials are advertisements that the advertiser wants to show visitors and can be a combination of both creative and text materials.

Depending on the nature of the advertisement campaign the advertiser wants to run, the size of the budgets can vary quite a lot. Further, some advertisers choose to define daily budgets (i.e., a daily spending limit) in order to control how the total of their budgets are being spent, e.g., to gain control of the length of an advertisement campaign. If an order has both a total and a daily budget, the data-model will consider the minimum of the two as the actual budget.

In our data-model we do not have direct access to the bids, instead we make decisions based on the eCPM predictions, which incorporate the bids.

### eCPM predictions

The ad exchange combines materials, bids, placements, time of day and other parameters to produce an eCPM lookup-table. At any time we can ask the ad exchange what the eCPM for a given material and placement combination is. The eCPMs are treated as a direct representation of publisher revenue for showing a material on a placement, and are used to decide what materials are shown for each impression.

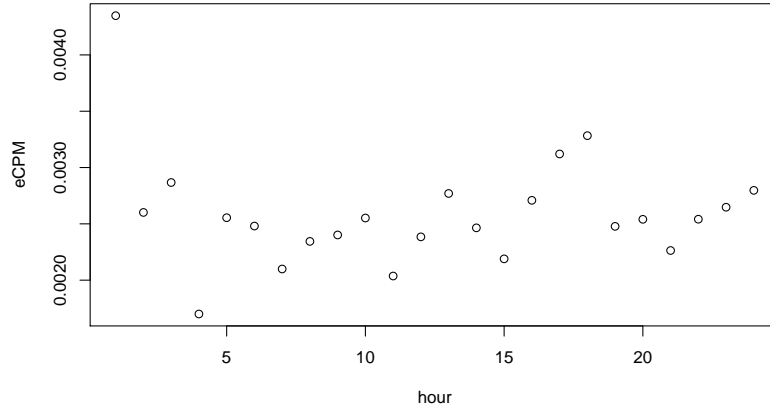
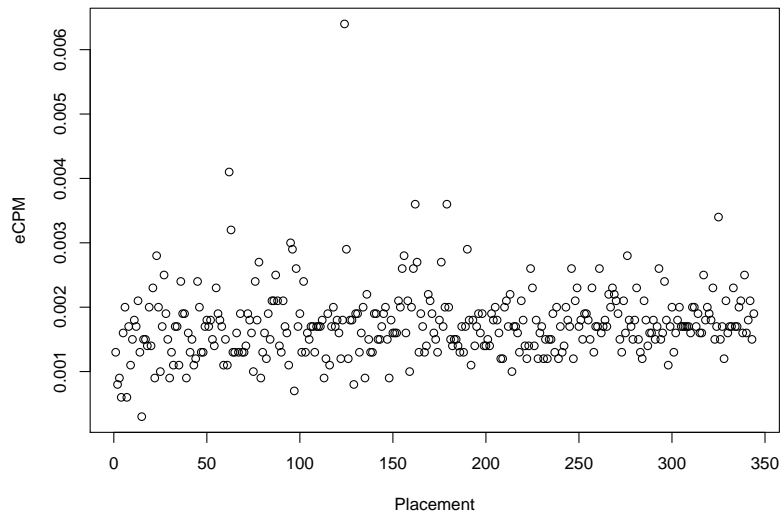
### Observed impressions

We have a set of observed impressions for each placement and time. In the on-line scenario this is undefined, but from recorded traffic we can read how many impressions each placement received during previous hours. These recorded impressions are ordered, so it is possible to trace exactly when each placement received impressions. This is useful in the Baseline algorithm, as it allows the algorithm to perform a *trace* of observed placement impressions, but substituting the real ad exchange with our own method of material selection.

### Data observations

Since the ad exchange operates in terms of eCPM, the behavior of eCPMs is of interest. eCPMs vary over both time and placement. As we can see in Figure 3.2 there can be fairly large fluctuations in the eCPM for an *order*  $\times$  *placement* combination over time. As the ad exchange greedily picks the highest eCPM material for each impression, it will not consider future time periods when the eCPM might be higher, as seen in the figure.

Further, Figure 3.3 illustrates that eCPM vary over placements for an *order*  $\times$  *time* combination. Since the publisher receives placement impressions, we need to decide for each impression if we should show a material, or save the budget for a later impression that is possibly on another placement when the eCPM might be higher.

FIGURE 3.2: eCPM over time for an  $order \times placement$  combination.FIGURE 3.3: eCPM over different placements for an  $order \times time$  combination.

## 3.2 Output data format

The output of the budget optimization methods presented in this thesis uses the same output data format. All methods render a  $placement \times material \times time$  impression allocation as seen in Table 3.1.

Using this output format we can read the number of impressions allocated to each  $placement \times material \times hour$  combination, and the revenue generated from doing so.

TABLE 3.1: An excerpt from a *placement*  $\times$  *material*  $\times$  *hour* impression allocation. The values in the placement and material columns are id-numbers.

Hour	Placement	Material	Assigned Impressions	eCPM
...				
9	42994	69368	4	0.00666033
9	51294	68882	3	0.00667524
9	53822	68882	7	0.0055314
...				

### 3.3 Case-study data

We have two publishers that we will use in the examples for the remainder of the thesis. One mid-size publisher MID and one large-size publisher LARGE.

We store the data on a daily basis (day-states), and aggregate impressions by hour. The data in each day-state is static, but since publishers can enable and disable placements, and orders enter and leave the system, the day-states can vary between days by more than just the received traffic.

When looking at the data and doing experiments we will be using days between the 12th of March 2012 to the 25th of March 2012. Due to some issues with the data, we have 13 days of data for MID and 11 days of data for publisher LARGE. All the tables in the next sections are averages over the respective day-states.

#### Publisher Mid

We present some descriptive statistics for the MID publisher.

TABLE 3.2: Impressions statistics for MID over the 13 days.

Measure	Impressions
Min	5,457,141
Max	10,525,559
Average	8,155,977

TABLE 3.3: Order count for MID over the 13 days.

Measure	Number of orders
Min	30
Max	45
Average	39

TABLE 3.4: Placement count for MID over the 13 days.

Measure	Number of placements
Min	5873
Max	7269
Average	6706

TABLE 3.5: Material count for MID over the 13 days.

Measure	Number of materials
Min	150
Max	247
Average	198

#### Publisher Large

The main difference between MID and LARGE is the amount of impressions received. LARGE has fewer placements and receives more traffic.

TABLE 3.6: Impressions statistics for LARGE over the 11 days.

Measure	Impressions
Min	21,687,735
Max	34,761,381
Average	29,005,436

TABLE 3.7: Order count for LARGE over the 11 days.

Measure	Number of orders
Min	82
Max	88
Average	86

TABLE 3.8: Placement count for LARGE over the 11 days.

Measure	Number of placements
Min	1,569
Max	1,717
Average	1,669

TABLE 3.9: Material count for LARGE over the 11 days.

Measure	Number of materials
Min	490
Max	611
Average	566

### 3.4 Summary

We have described the data format that will be used as input for the Baseline in Chapter 4, the off-line optimization model in Chapter 5, the traffic prediction in Chapter 6 and the on-line optimization heuristic in Chapter 7.

We have seen how eCPM changes over both time and placement. The observations implies that there exists potential for optimizing how materials are being selected, since the current ad exchange is unable to consider higher eCPM impressions in the future.

We have shown real data for two publishers, MID and LARGE. MID and LARGE will be used in the experiments for the rest of this thesis.

## A formal Baseline algorithm

In the previous chapter we defined the input data model. This chapter will present a greedy Baseline algorithm that simulates a real ad exchange. We will see that the Baseline algorithm is unable to consider high eCPM impressions arriving late in the day due to over-spending of order budgets early in the day. This motivates the development of a budget optimization model.

The revenue of the Baseline algorithm will be used as a revenue lower bound.

### 4.1 An ad exchange algorithm

We want our Baseline algorithm to parse impressions in a greedy on-line fashion, and for each placement impression pick the highest eCPM material  $m$ , if the order containing  $m$  has enough budget left.

We have developed a simple, greedy algorithm that illustrates the behaviour of the ad exchange. It is defined in Algorithm 1.

---

#### Algorithm 1 Ad exchange algorithm

---

```

I := stream of placement impressions
eCPMm,i := revenue of showing material m for impression i
M := Set of materials.
order(m) := remaining budget for order containing material m
for all i ∈ I do
    sort(M) w.r.t. eCPMm∈M,i
    for all m ∈ M do
        if order(m) − eCPMm,i ≥ 0 then
            show material m for this impression
            reduce order(m) by eCPMm,i
            break m ∈ M loop
        end if
    end for
end for

```

---

Algorithm 1 sorts the materials by eCPM highest to lowest for each placement, and picks the first material with enough budget left. While this serves to give an easy to grasp representation of the problem, it does not consider text materials, and it is not adapted to our data model format.

## 4.2 The Baseline algorithm

Algorithm 1 picks the highest eCPM material  $m$  as long as the order of  $m$  has budget left. It parses all impressions in sequence.

However, we need more domain constraints in our Baseline to be able to parse the data we have available. Most importantly, we need to be able to consider creative materials as well as text materials. Adding these considerations will further enhance the relevance of the results of the Baseline as well. Recall from Chapter 3 that our eCPM predictions and placement impressions are aggregated by hour. The ordering of the impressions is preserved, but they are divided into a separate set for each time-period in the system.

We adapt Algorithm 1 to consider these additions in Algorithm 2.

---

**Algorithm 2** The Baseline algorithm.

---

$T$  := ordered set of time-periods.  
 $I_t$  := stream of ordered placement impressions divided into hours  $t \in T$ .  
 $placement(i)$  := placement for impression  $i$ .  
 $P$  := set of placements.  
 $eCPM_{t,p,m}$  := revenue of showing material  $m$  on placement  $p$  at time  $t$ .  
 $order(m)$  := remaining budget for order containing material  $m$ .  
 $\mathcal{C}$  := set of creative materials.  
 $\mathcal{T}$  := set of text materials.  
 $\mathbb{C}_p$  := text material capacity for placement  $p$ .

```

for all  $t \in T$  do
  for all  $p \in P$  do
     $\mathcal{C}_p \leftarrow \text{sort } \mathcal{C} \text{ w.r.t. } eCPM_{t,p,m}$ 
     $\mathcal{T}_p \leftarrow \text{sort } \mathcal{T} \text{ w.r.t. } eCPM_{t,p,m}$ 
  end for
  for all  $i \in I_t$  do
     $p \leftarrow \text{placement}(i)$ 
     $c \leftarrow \text{takeWithBudgetLeft}(\mathcal{C}_p, 1)$ 
     $ts \leftarrow \text{takeWithBudgetLeft}(\mathcal{T}_p, \mathbb{C}_p)$ 
    if  $eCPM_{t,p,c} \geq \sum_{m \in ts} eCPM_{t,p,m}$  then
      assign  $c$  to  $i$ 
       $order(c) \leftarrow order(c) - eCPM_{t,p,c}$ 
    else
      for all  $m \in ts$  do
        assign  $m$  to  $i$ 
         $order(m) \leftarrow order(m) - eCPM_{t,p,m}$ 
      end for
    end if
  end for
end for

```

---

Algorithm 2 looks at each time period, and sorts the materials *per placement* by their eCPMs, highest to lowest. The function *takeWithBudgetLeft* takes an ordered set of materials  $M$  and an integer  $n$  as input; it returns the first available set  $s \subset M$  where  $|s| = n$  and each element of  $s$  is unique; and each



$order(m)$  for all  $m \in s$  have sufficient budget left to be charged the current eCPM of  $m$ .

The algorithm parses the stream of impressions  $I_t$ , for each hour it picks the best creative material  $c$ , and a set of the highest eCPM text materials  $ts$ . The algorithm then picks either material  $c$  or the set of materials  $ts$  as the “winner” for the impression  $i$  by which has the highest eCPM total. The winner(s) are charged the eCPM to their budget, and the procedure is repeated until all impressions are parsed.

### 4.3 Comparison to production

The Baseline (Algorithm 2) is not an exact match of what is used in a production-level ad exchange. There are a significant amount of constraints that the production algorithm considers that the Baseline does not. However, it does capture the most relevant aspects, and comparisons with the production system show that the revenue figures produced by the Baseline are in the same “ball-park” as the actual revenue that was realized.

### 4.4 Baseline budget distribution and performance

To illustrate how the Baseline performs, and further motivate the need for a budget optimization model, we present a graph seen in Figure 4.1 that shows how the Baseline distributes the budgets over time for two example orders belonging to MID.

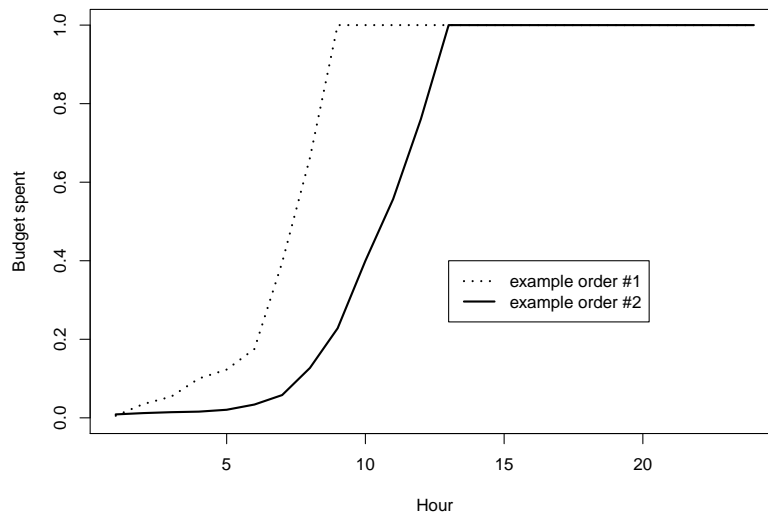


FIGURE 4.1: Budget utilization by the Baseline algorithm for two example orders. The orders belong to MID.

It is clear that the Baseline algorithm has a strong tendency to aggressively spend the budgets of the orders early in the day. By consuming the budgets too early, the Baseline is unable to consider high eCPM impressions that arrive late in the day. We know these kind of impressions exist, as seen in Figure 3.2. By implementing a budget optimization model in the following chapters we can allocate impressions more efficiently, rendering higher publisher revenue.

## 4.5 Summary

We have presented a formal greedy Baseline algorithm that simulates how materials are selected in a real ad exchange. The output of the Baseline serves to provide a measurement of the increase in revenue of both the off-line budget optimization and the on-line budget optimization.

We further illustrated that the Baseline algorithm utilizes the budgets of orders in an aggressive, and seemingly sub-optimal fashion, since the algorithm is unable to consider high eCPM impressions arriving late in the day.

In the next chapter we will look at a linear programming model for solving the budget optimization problem in an off-line setting.

## Off-line budget optimization

This chapter describes a *linear programming* (LP) model for doing off-line budget optimization.

The LP will provide a formal definition of the problem we are trying to solve. The solution of the LP will provide a revenue *upper bound*. We know that the output of an LP is optimal in the context of the model, hence we have no hope of achieving higher revenue than the LP model.

The use of linear programming for solving the off-line problem is not unheard of [CBAD11, FIMN08].

### 5.1 Linear programming

Linear programming [Chv83] is a general method of formulating mathematical optimization problems. By defining an objective function that we either want to maximize or minimize the value of; combined with a set of linear (in-)equality constraints, an LP solver can traverse the space of feasible solutions and locate an optimal solution, if one exists. The general definition of an LP is

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \leq b \\ \text{and} & x \geq 0 \end{array}$$

In this case  $x$  is the *decision variable* we wish to find an optimal assignment to; and  $c$ ,  $b$  and  $A$  are parameters to the model.

Note that the LP does not require  $x$  to be integer. In our case, when  $x$  has the value of impressions we cannot possibly assign a fraction of an impression to a placement. However, the *amount* of impressions that we are working with makes the error from the relaxation insignificant. For example, if we assign 5,000 impressions or 5,000.4 impressions to a placement is of little significance when dealing with millions of impressions in total.

### 5.2 Model input

The model we develop aggregates impressions by hour. We also choose to work with “days”, i.e., 24 hour periods. Both these choices are easily manipulated, and the granularity of the impression aggregation and how long time-periods you work with is bounded only by the ever increasing combinatorial difficulty of the problem.

The parameters of the LP model the model are

- The observed impressions per placement for a given publisher.
- Orders: orders are a set of materials, and a budget.
- Materials: a material can be either *text* or *creative*. A text material is a simple piece of plain text, and a creative is either an image or an Adobe Flash script.
- Placements: each placement can either only accept creative materials or text materials, or both. If a placement accepts text materials, it has a *text material capacity* defining how many text materials may fit on the same placement. This makes text materials special, since many text materials may share the same placement impression. We let placements that do not accept text materials have a text capacity of zero.
- An eCPM prediction model; rendering eCPM values for every *hour*  $\times$  *placement*  $\times$  *material* combination. We can use the eCPM values as a direct representation of publisher revenue.

### 5.3 Objective

Our model has one decision variable  $x$ .  $x$  is a three-dimensional matrix, and each element of  $x$  is a value for how many impressions are allocated to each *hour*  $\times$  *placement*  $\times$  *material* combination.

Let  $T$  be the set of hours,  $\mathcal{M}$  the set of materials,  $\mathcal{P}$  the set of placements and  $v_{t,p,m}$  the eCPM for any combination of  $t \in T, p \in \mathcal{P}, m \in \mathcal{M}$ . The objective function for maximizing revenue is then formulated as

$$\text{maximize } \sum_{t \in T} \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} x_{t,p,m} \cdot v_{t,p,m} \quad (5.1)$$

### 5.4 Constraints

- Let  $I_{t,p}$  be the amount of impressions that placement  $p \in \mathcal{P}$  will receive at hour  $t \in T$ .
- Let an order  $o \in \mathcal{O}$  represent the materials belonging to that order.
- Let  $\mathcal{B}_o$  be the budget for order  $o$  in the set of all orders  $\mathcal{O}$ .

#### Constraints for creative materials

To better illustrate the core constraints of the model, we consider the case when the only type of materials are creative materials. Having only creative materials means we can fit exactly one material per placement. The constraints are intuitively formulated as

$$\forall o \in \mathcal{O}. \sum_{t \in T} \sum_{p \in \mathcal{P}} \sum_{m \in o} x_{t,p,m} \cdot v_{t,p,m} \leq \mathcal{B}_o \quad (5.2)$$

(5.2) constrains budget consumption to be equal to or less than the budget limit of each order.

We also need to constrain the material assignment to the number of available impressions, we do this in (5.3).

$$\forall t \in T. \forall p \in \mathcal{P}. \sum_{m \in \mathcal{M}} x_{t,p,m} \leq I_{t,p} \quad (5.3)$$

### Constraints for both creative and text materials

The constraints (5.2) and (5.3) do not consider text materials. Text materials complicate the model in the following ways:

1. Many text materials can fit on the same placement. Since we count placement impressions, text materials consume fractions of impressions.
2. Different placements can fit a different number of text materials.
3. We can only use a distinct text material once per placement impression.

To make the model consider these facts, we define two intermediate decision variables with one constraint each. Let  $\mathcal{C}$  be the set of all creative materials, and  $\mathcal{T}$  the set of all text materials.

$$\forall p \in \mathcal{P}. \forall t \in T. \alpha_{p,t} = \sum_{m \in \mathcal{C}} x_{t,p,m} \quad (5.4)$$

$$\forall p \in \mathcal{P}. \forall t \in T. \beta_{p,t} = \sum_{m \in \mathcal{T}} x_{t,p,m} \quad (5.5)$$

$\alpha_{p,t}$  is the number of impressions used by creative materials for placement  $p$  at time  $t$ . Respectively,  $\beta_{p,t}$  is the number of impressions used by text materials. To make sure a text material  $m$  appear at most once for each impression, we constrain each individual text material to the number of available impressions not consumed by creative materials.

$$\forall t \in T. \forall p \in \mathcal{P}. \sum_{m \in \mathcal{T}} x_{t,p,m} \leq I_{t,p} - \alpha_{p,t} \quad (5.6)$$

We can now constrain the sum of  $\alpha$  and  $\beta$  to be less than the number of available impressions. Let  $\mathbb{C}_p$  be the text material capacity for placement  $p$ .

$$\forall t \in T. \forall p \in \mathcal{P}. \alpha_{t,p} + \frac{\beta_{t,p}}{\mathbb{C}_p} \leq I_{t,p} \quad (5.7)$$

## 5.5 Baseline comparison

We can run the LP using observed traffic, and compare to the revenue of running the Baseline on the same data.

TABLE 5.1: LP improvement for MID and LARGE compared to the Baseline algorithm.

Measure	MID improvement	LARGE improvement
Max	7.77%	8.43%
Min	5.34%	2.98%
Average	6.67%	6.55%

## Budget distribution improvement

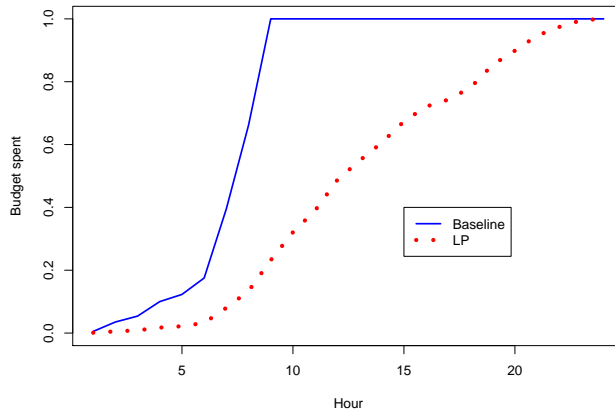


FIGURE 5.1: Comparison of how the LP spends the budget of example order #1 compared to the Baseline algorithm.

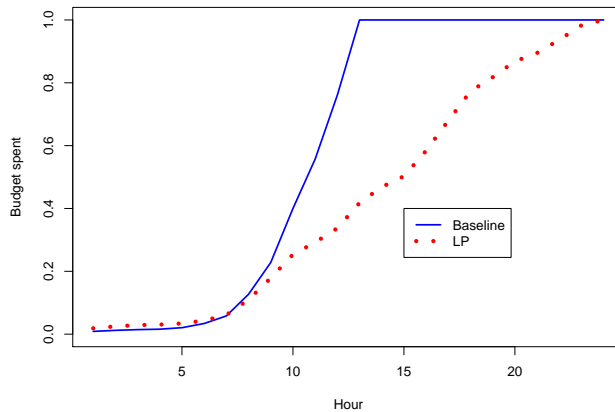


FIGURE 5.2: Comparison of how the LP spends the budget of example order #2 compared to the Baseline algorithm.

Figures 5.1 and 5.2 shows how the LP model allocates the budget of the two example orders used in Chapter 4.

As can be seen, the LP consumes the budget of the orders evenly through the day, taking maximum advantage of each impression. By maximizing the utility of each impression the LP ensures that orders will consume their budgets using as few impressions as possible. We will further analyze this behavior in Chapter 7.

## 5.6 Budget limit impacts

The amount of orders consuming the total of their budget within the period solved by the LP affects the potential for optimization. We illustrate this by cutting the budgets of the orders. For MID we cap the budgets of the orders to 4000, 5000 and 7500 units of revenue respectively.

TABLE 5.2: Average improvement compared the Baseline paired with the amount of orders consuming the total of their budget.

Budget cap	Orders consuming budget	Improvement
4000	39.24%	8.20%
5000	35.28%	7.51%
7500	33.21%	6.71%
Actual	34.86%	6.67%

Table 5.2 shows that the amount of orders completely consuming their budgets have a direct impact on the optimization potential. It is important to note that a high degree of optimization is not necessarily desirable in itself. In the best of cases all orders have an unlimited budget, allowing us to freely pick the highest eCPM materials for each placement without any constraints.

## 5.7 Additional model constraints

While our LP model does define the basic problem of material selection, there are many domain-details that it does not consider. Primarily there is an issue with *homepage takeover*. Consider a case where an advertiser has high eCPM on many placements. This means that the advertiser may appear on many or all of the placements available on a webpage - which from the perspective of the publisher is not desirable, since it will appear as if the advertiser has “taken over” the web page.

Another issue with the LP is that it may not scale well to added domain-details. It works well within the context of this thesis, but adding more and more constraints and details will most likely make the run-time an issue. And it will also be harder to verify the correctness of the model since errors in the model description are more likely. Hence, for use in production, an alternative to the LP may be desirable.

## 5.8 Summary

We have shown an LP model that can render a 6% increase in publisher revenue for off-line data, compared to the Baseline algorithm; and that this figure depends on how many orders completely consume their budgets. This proves that there is potential for optimization in the on-line scenario.

To be able to utilize the LP in an on-line setting, the next chapter will look at a linear regression model for predicting future placement impressions.

## Traffic prediction

In this chapter we will discuss the development of a placement impression prediction model that can be used by the LP-model from the previous chapter. We will use linear regression to fit a model that captures both weekly and daily trends.

Figure 6.1 illustrates the data we are trying to predict.

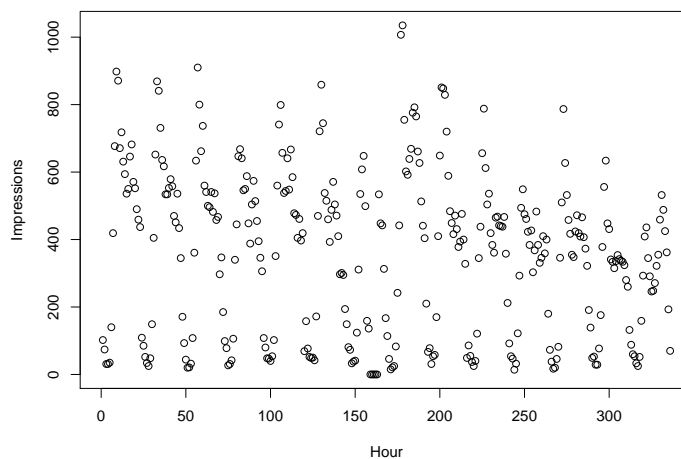


FIGURE 6.1: Historic impressions for a placement belonging to MID. We observe clear trends where the placement consistently receives less impression during nights and early mornings, and peaks during noon.

### 6.1 Problem definition

The traffic prediction has the following input available:

- A set of placements  $P = \{p_1, \dots, p_n\}$
- Historic traffic  $y_{p_i}(t)$  for each placement  $p_i \in P$  and hour  $t$ .

It is the case that the number of impressions that a placement receives is much greater during days than night, i.e., for each placement we have *high*



*traffic hours* and *low traffic hours*. Since the high traffic hours constitute a large majority of all impressions received, we state that it is more important for our traffic prediction model to perform well at the high traffic hours than at the low traffic hours. We formally define high traffic hours as: an hour  $t$  is high traffic for a placement  $p$  if and only if

$$y_p(t) \geq \text{mean}(y_p)$$

Defined as such, the high traffic hours represent roughly 80% of the total amount of impressions.

Further, we are interested in predicting impression amounts per hour. A higher granularity would not be beneficial in the context of this thesis as the LP-model in Chapter 5 considers an hour to be the smallest unit of time. Moreover, as the LP runs on 24 hour periods, we are also happy if we can predict 24 consecutive hours.

Our goal is to find a function  $f_p(t)$  that for future hours  $t$  will give a prediction for the amount of impressions placement  $p$  will receive at hour  $t$ .

For more information about linear regression see [MA02].

## 6.2 Input data

To develop the model, we have decided to look at the *top impression receiving placements*.

As seen in Figure 6.2 it is the case that a few of the placements receive a large portion of the total impressions (top banners etc.). Hence, we have selected the top ten placements for MID and the top eleven placements for LARGE. For these placements we have six months worth of recorded data, or 4416 hours in total. Later in the chapter when we present evaluations of our model, we will present averages over three predictions, using 16%, 50% and 83% of the data for training and the remainder of the data for evaluation.

The traffic periods used in the majority of the experiments in this chapter range from August 2011 to January 2012. We then take the best model and use it with our case-study data.

## 6.3 Performance measurements

How to get a good measure of the performance of our model is not obvious. The value we are most interested in is the *mean percentage error*, or MPE. Let  $e_t$  be the expected (predicted) amount of impressions for hour  $t$ , and  $o_t$  be the observed amount of impressions for hour  $t$ . For a set  $T$  of predicted hours, MPE is defined as  $\frac{1}{|T|} \cdot \sum_{t \in T} \frac{|e_t - o_t|}{o_t}$ . In the current context, there is an issue with the MPE measure: Assume we have a prediction that always expects one extra impression than what is observed. During high traffic hours the error will be negligible, however, during low-traffic hours the percentile error of even one impression can be very large (e.g. we observe one impression and we expect two, this constitutes an error of 100%) even though the significance of that error is irrelevant from an economic perspective.

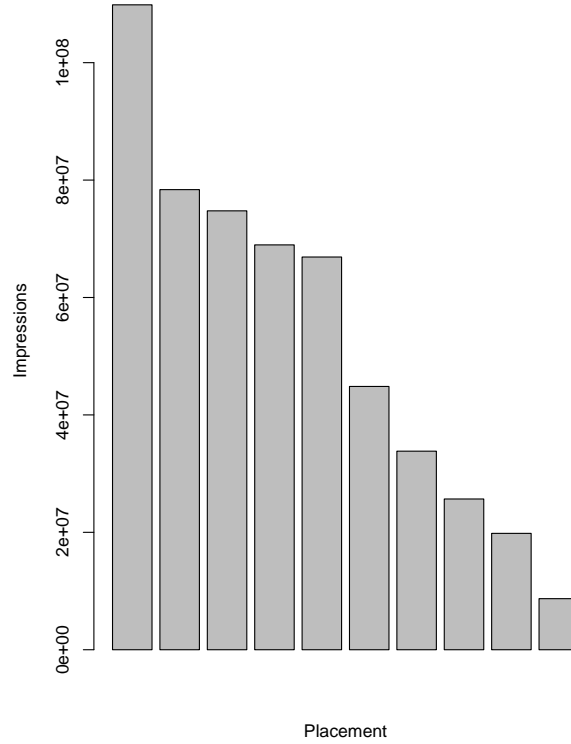


FIGURE 6.2: Total amount of impressions for the ten most popular placements for MID.

We have thus devised a *weighted MPE* that weights the percentile error of a predicted hour with the economic significance of that hour. We define weighted MPE in (6.1).

$$\sum_{t \in T} \left( \frac{|e_t - o_t|}{o_t} \cdot \frac{o_t}{\sum_{t' \in T} o_{t'}} \right) \quad (6.1)$$

For the remainder of this chapter, when we measure the performance of our model, we will provide MPE and weighted MPE measures both for complete 24 hour periods, and when we measure high traffic hours.

## 6.4 Weekly trends

To get a better understanding of how the data behaves over longer periods of time, we have done weekly trend analysis by looking at how many impressions a given hour  $t$  and placement  $p$  receives over weekdays (e.g. Mondays). In Figure 6.3 we do a trend plot that looks at how the traffic varies over weekdays<sup>1</sup>.

<sup>1</sup>168 is the amount of hours in a week.

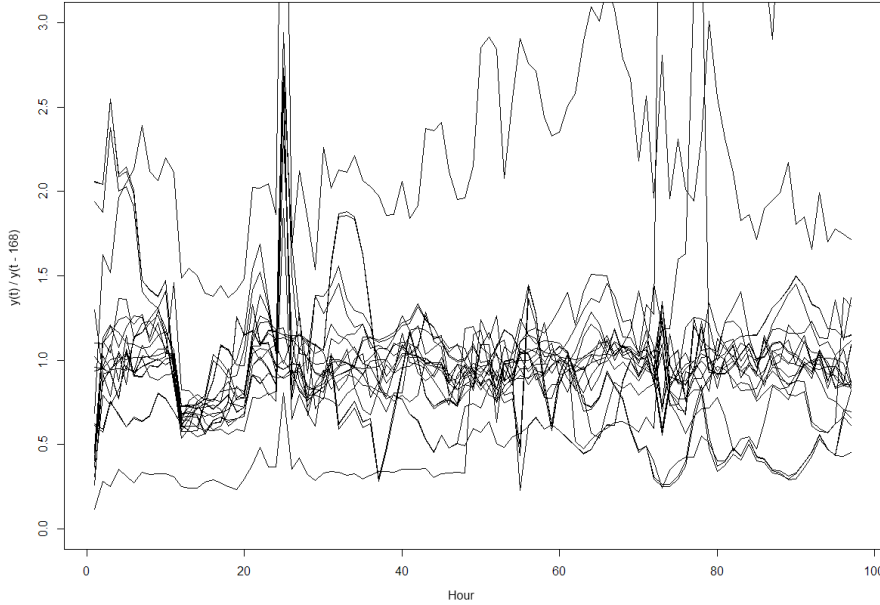


FIGURE 6.3:  $\frac{y(t)}{y(t-168)}$  plotted for a set of placements belonging to MID.

Even if the trend looks a bit noisy, it is still mostly linear, so we decided to do a linear fit to capture the weekly trends. We call this function  $f_{week}(t)$ , and it fits four parameters  $\alpha_1, \alpha_2, \alpha_3$  and  $\alpha_4$ .

$$f_{week}(t) = \alpha_1 + \alpha_2 \cdot y(t - 168) + \alpha_3 \cdot y(t - 336) + \alpha_4 \cdot y(t - 504) \quad (6.2)$$

### Week model performance

TABLE 6.1: MPE measurements for  $f_{week}$  predictions.

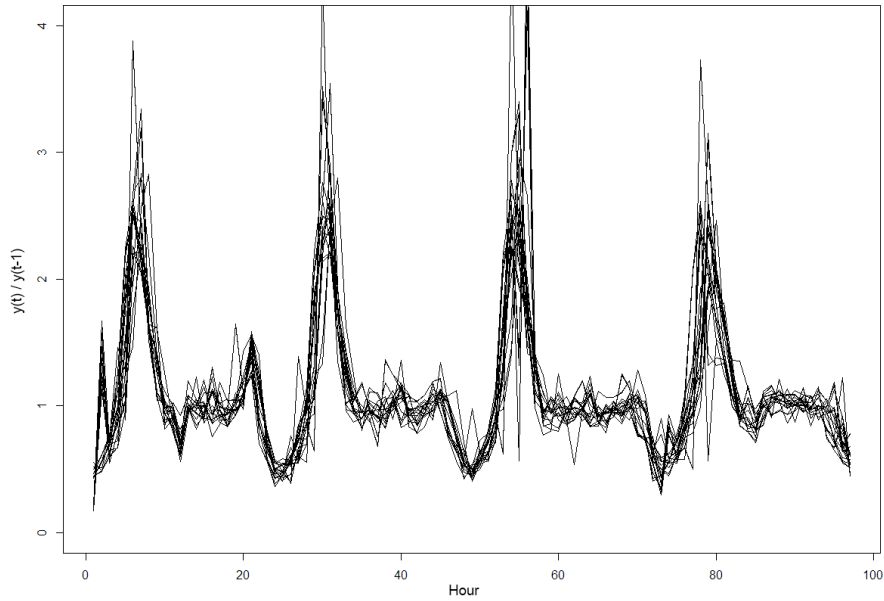
Data	Average MPE	Max MPE	Min MPE
MPE	45.27%	117.64%	12.10%
Weighted errors	24.45%	46.59%	8.69%
high traffic	19.72%	38.69%	7.34%
Weighted high traffic	20.07%	38.25%	7.11%

The MPE measurements from using only  $f_{week}$  to predict traffic can be seen in Table 6.1. To put them in perspective, we can use a naive prediction method of simply copying hour  $t - 24$  as the prediction for hour  $t$  (copying the previous day).

What we can see in Table 6.2 is that while our model  $f_{week}$  performs worse if measured over all hours, it is actually the case that we handle the important hours quite a lot better.

TABLE 6.2: Performance of  $f_{week}$  compared to copying the previous days traffic as a prediction.

Data	Average MPE	Max MPE	Min MPE
MPE	-36.31%	-73.33%	20.76%
Weighted	8.70%	7.04%	26.42%
high traffic	15.83%	9.69%	29.36%
Weighted high traffic	14.34%	10.46%	32.99%

FIGURE 6.4:  $\frac{y(t)}{y(t-1)}$  plotted for a set of placements belonging to MID.

## 6.5 Hour model

Even though we desire to be able to predict a complete day, to get a better understanding of how the traffic changes during the day, we developed an hour prediction model.

We started by doing the same type of trend analysis, as can be seen in Figure 6.4.

From the analysis we can see that during mornings and afternoons there is a clear exponential trend, while it stays fairly linear during the middle of the day and during the night.

The model  $f_{week}$  will capture the linear trend, and we will add an exponential model to fit the residue from  $f_{week}$ .

Let  $\hat{y}$  be the residue from the  $f_{week}$  model,  $\hat{y}(t) = y(t) - f_{week}(t)$ . We want to fit an exponential model to  $\hat{y}$ .

$$f_{hour}(t) = \beta_1 \cdot e^{\beta_2 \cdot \log \hat{y}(t-1)} \quad (6.3)$$

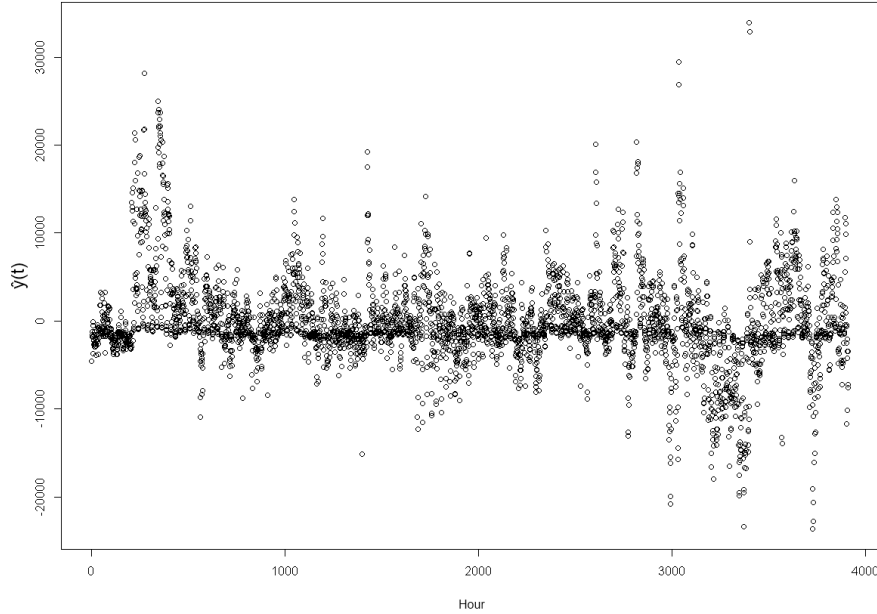


FIGURE 6.5: Weekly prediction residue.

TABLE 6.3: The measured MPEs for the hour prediction model.

Data	Average MPE	Max MPE	Min MPE
All hours	30.88%	73.52%	9.26%
Weighted all hours	16.02%	29.44%	6.29%
high traffic	12.8%	23.27%	5.42%
Weighted high traffic	12.96%	24.22%	5.4%

TABLE 6.4: Hour model improvement compared to using hour  $t - 1$  as prediction for hour  $t$ .

Data	Average MPE	Max MPE	Min MPE
All hours	4.46%	-29.44%	60.12%
Weighted errors, all hours	13.78%	-23.54%	56.92%
high traffic hours	4.83%	-17.58%	40.11%
Weight errors, high traffic hours	0.99%	-27.81%	39.33%

and we then have a complete model for predicting the next hour  $t + 1$  as

$$f(t) = f_{week}(t) + f_{hour}(t) \quad (6.4)$$

### Hour model performance

As the benchmark for the hour prediction model, as it only predicts the next hour, we copy the previous hour and use it as the prediction for the next hour.

By looking at the slight improvements we see that copying the previous hour as a prediction is actually a viable solution if we *only* care about the high traffic hours and we only want to predict the next immediate hour.

## 6.6 Day prediction model

As we saw in the previous section, using an exponential fit is something that works fairly well. For high traffic hours the weighted MPE is only around 10-15% which is good enough for our cases. However, we need to be able to predict a continuous 24 hour period.

Again, we turn to trend analysis as seen in Figure 6.6.

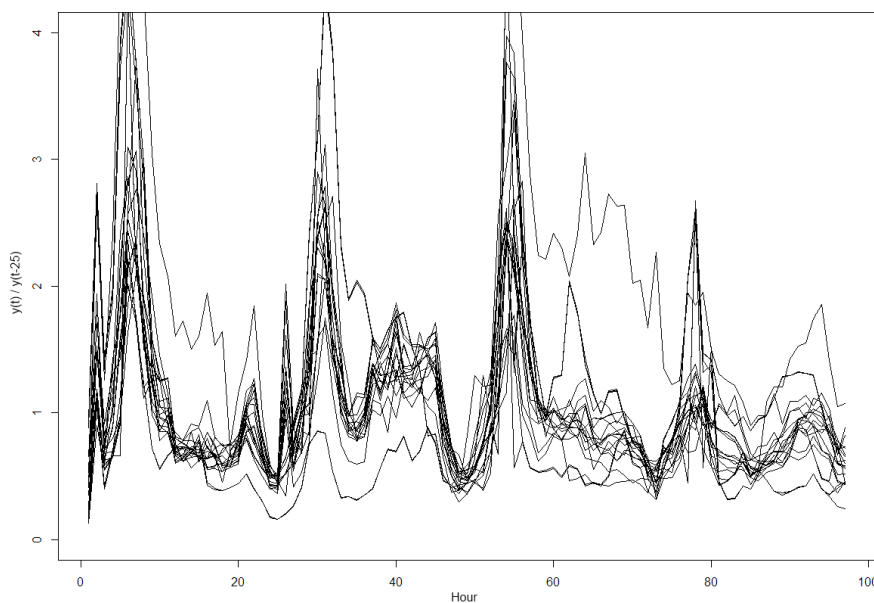


FIGURE 6.6: Trend analysis on  $\frac{y(t)}{y(t-25)}$ .

While the trend is not as clear as in Figure 6.4, we still see rapid changes in traffic volumes during mornings and evenings. To not risk over-reacting to these changes, we change from an exponential fit to a second order fit. Since the trend is noisier we also opt to use more data-points from the previous day. Experiments have shown that the following works well. The vectors  $\beta$  and  $\gamma$  are fitted to the training data.

$$\begin{aligned}
 f_{day}(t) = & \beta_1 + \beta_2 \cdot \hat{y}(t-24)^2 + \gamma_1 \cdot \hat{y}(t-24) + \\
 & \beta_3 \cdot \hat{y}(t-25)^2 + \gamma_2 \cdot \hat{y}(t-25) + \\
 & \beta_4 \cdot \hat{y}(t-26)^2 + \gamma_3 \cdot \hat{y}(t-26) + \\
 & \beta_5 \cdot \hat{y}(t-27)^2 + \gamma_4 \cdot \hat{y}(t-27)
 \end{aligned}$$

The complete day prediction model becomes

$$f(t) = f_{week}(t) + f_{day}(t)$$

### Day model performance

To evaluate the day prediction model, we return to the benchmark used to evaluate  $f_{week}$  and copy the previous days traffic ( $t - 24$ ) as the prediction, the results are seen in Table 6.5.

TABLE 6.5: Day prediction performance compared to using yesterdays traffic as a prediction.

Data	Average MPE	Max MPE	Min MPE
All hours	-22.84%	-59.89%	9.63%
Weighted all hours	11.47%	8.03%	17.14%
high traffic	16.37%	12.07%	24.33%
Weighted high traffic	15.1%	12.46%	28.23%

## 6.7 Results

All the measurements presented above were using the same six months of data, 21 placements and three predictions for each of the placements using different amounts of the data for training. The final results were averaged over the three predictions.

By doing some manual tweaking and experimenting, we found that using two months of training data rendered the best results on our data set.

TABLE 6.6: Performance of the day prediction model using two months of data for training compared to using yesterdays traffic as a prediction.

Data	Average MPE	Max MPE	Min MPE
All hours	-19.98%	-44.98%	17.22%
Weighted errors, all hours	16.21%	13.85%	16.09%
High traffic	21.08%	21.28%	17.94%
Weighted errors, high traffic	19.41%	15.53%	18.63%

As we can see in Table 6.6, for the high traffic hours and with the significance of the predicted hour considered we achieve an improvement of almost 20% compared to the very naive method of just copying historic traffic. Figure 6.7 shows two example predictions.

## 6.8 Results on case-study data

There is an issue with doing the traffic prediction for a large set of placements: The publisher may add or remove placements from the system at will. The traffic pattern in Figure 6.1 is fairly regular, but we also have more irregular patterns as seen in Figure 6.8.

To decide which placements we predict traffic for, we picked the top impression receiving placements for both MID and LARGE, a set constituting about 90% of the total received impressions. By looking at the traffic these placements received we removed the placements with incomplete traffic history from the set. Another method to handle incomplete history would have been to copy hour  $t - 24$  as a prediction for hour  $t$ . Since this chapter is about evaluating the traffic prediction model, we removed the placements instead.

## 6.8. Results on case-study data

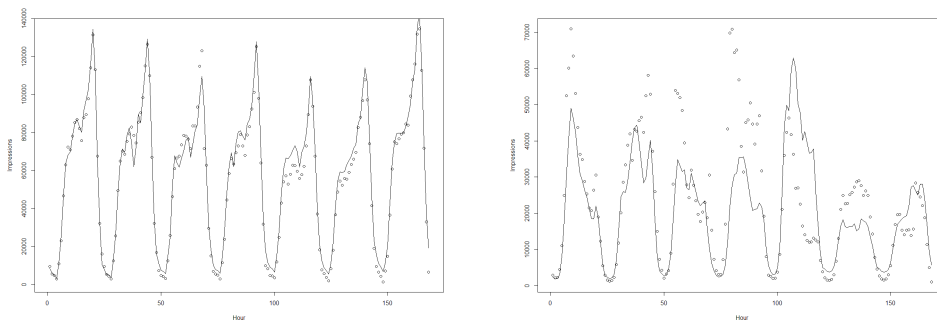


FIGURE 6.7: Two examples of predicted traffic. The solid line represents expected impressions and circles are observed impressions.

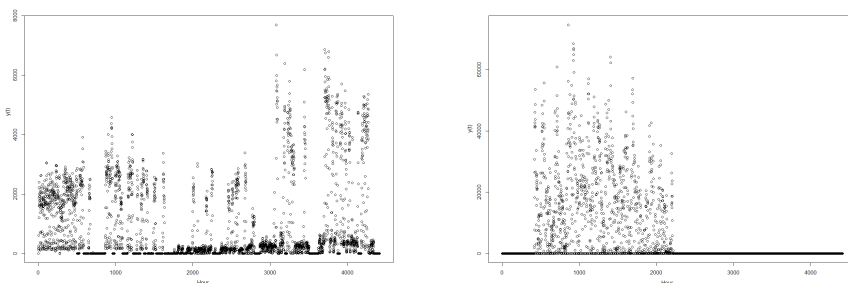


FIGURE 6.8: Problematic traffic patterns.

After filtering placements that have incomplete training data, the amount of placements we predict is presented in Table 6.7. To be able to fit as many placements as possible only one month of training data were used.

TABLE 6.7: Number of placement that were predicted for each of the publishers.

Publisher	Placements
MID	114
LARGE	132

TABLE 6.8: Measured traffic prediction errors for the Case Study data.

Publisher & hours	Avg Weighted MPE	Max Weighted MPE	Min Weighted MPE
MID All hours	30.5%	75.8%	9.2%
MID High traffic hours	25.7%	68.7%	7.5%
LARGE All hours	39.0%	513.1%	4.1%
LARGE high traffic hours	29.3%	221.1%	3.2%

Measuring the improvement for the errors compared to copying previous day as prediction.

As can be seen in Table 6.9 when we increase the number of placements that we want to predict combined with having less training data, it is hard to out-perform copying the previous day as a prediction.



TABLE 6.9: Comparing the Weighted MPE for the Case Study prediction with copying the previous day as prediction.

Publisher & hours	Avg Weighted MPE Improvement
MID All hours	-1.279%
MID High traffic hours	1.797%
LARGE All hours	-4.206%
LARGE high traffic hours	11.049%

## 6.9 Summary

Even though we can gain a bit of accuracy with our model, as long as the placements are well-behaved our model has the potential to render up to 20% better accuracy compared to copying historic data as a prediction.

But when we look at *all* placements, we see that the potential of 20% increase diminishes, due to erratic placement behaviour and plain differences in traffic patterns between placements. While the results are good for high traffic hours on the placements with most traffic, the prediction performs poorly in other scenarios. The fact of the matter is, that the benchmark method used in this section requires no training data except the previous day, and it has the potential to render almost as good or better results when the traffic is low or more erratic.

As we shall see in the next chapter, the quality of the traffic prediction is of importance, and a more sophisticated and robust method may be desirable.

## On-line budget optimization

From Chapter 1, recall the figure of the budget optimization system, also shown here. We now have the necessary input available to define the on-line budget optimization heuristic. We define the “ROI-Finder” component in Figure 7.1 as measuring a *return-on-investment* (ROI) value for orders totally consuming their budgets in some *placement*  $\times$  *material*  $\times$  *time* impression allocation  $S$ , and how we can use a matrix of such ROI values to adjust the eCPM of orders, to render higher revenue than the Baseline algorithm.

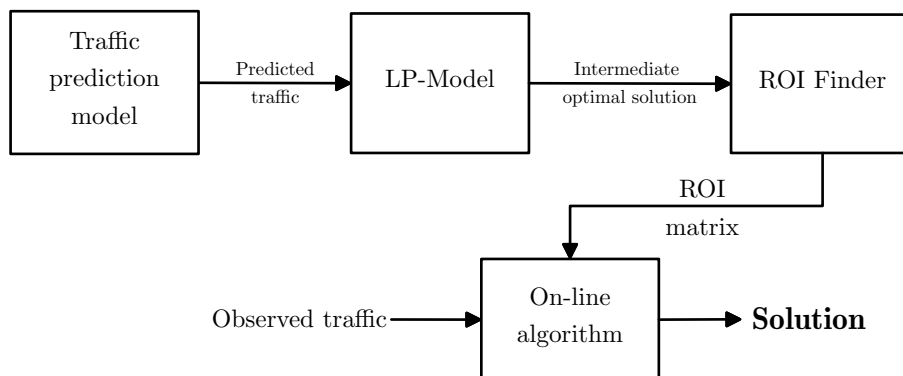


FIGURE 7.1: The budget optimization model.

The assumption of having an allocation  $S$  is well motivated, as from Chapter 5 we have access to an optimal allocation for observed traffic. Combined with the results from Chapter 6 we can also get an optimal allocation for predicted traffic.

The purpose of this chapter is to present the on-line budget optimization heuristic, combining the results of Chapter 5 and 6 and comparing the results of the budget optimization model with the Baseline algorithm presented in Chapter 4.

### 7.1 Problem definition

*We want to find an on-line heuristic that renders optimal, or closer to optimal results compared to the Baseline algorithm presented in Chapter 4.*

Even with an optimal allocation  $S$  in hand, the information available to us is limited.  $S$  simply provides us with a static schedule of when, where and how many times to show a material. Such a static schedule is not flexible enough to be useful in practice.

As we have seen in Chapter 6 placements are added and removed, and traffic predictions are far from perfect. Furthermore, in the production system, orders can enter and leave the system and bids can be changed. In this context, our heuristic must be flexible enough to accommodate unforeseen changes still render higher revenue than the Baseline.

The static allocation provided by  $S$  gives us some insight in how to do material selection, by looking at the difference in eCPM between the material that won impressions and the materials that did not.

## 7.2 Order classification

In the case that orders have unlimited, or infinite, budgets, it is always the case that for each impression we want to pick the highest eCPM materials since we do not have any budget constraints.

On the other hand, if all orders were to consume their budgets, there is no further revenue to realize, as the resources are depleted.

Hence, the interesting case is when the system is in such a state that *some* of the orders will consume their budgets and some will not. This allows the budget optimization model to reduce the number of impressions the orders that consume their budgets use, to *free up* impressions for the orders that have budget left.

We classify orders as either *budgeted* or *non-budgeted* depending on if the order consumes its budget in  $S$ .

## 7.3 Return on investment

As stated in the previous section, we have budgeted and non-budgeted orders. In this context, if a non-budgeted order  $P$  has the highest eCPM, it is *always* the optimal choice to let  $P$  win the impression.

In the case that a budgeted order  $O$  has the highest eCPM, we know that it may potentially be the case that  $O$  is better saved until a later impression, so as to consume  $O$ 's budget in a more optimal fashion. In this sense, we can consider non-budgeted orders *free of charge*, as they are always optimal to show if they have the highest eCPM. Budgeted orders on the other hand should “motivate” that they are some measure “better” to choose now, rather than save their budget for a later, alternative impression. We call this measure *return on investment* (ROI). Only budgeted orders have ROI values, and it is possible to measure ROI values from any *placement*  $\times$  *material*  $\times$  *time* allocation  $S$ .

For some budgeted order  $\mathcal{O}$ , placement  $p$  and time  $t$  where  $\mathcal{O}$  has won an impression, let  $\mathcal{P}$  be the non-budgeted order with the highest eCPM (among non-budgeted orders)

$$ROI_{t,p}(\mathcal{O}) = \frac{eCPM_{t,p}(\mathcal{O}) - eCPM_{t,p}(\mathcal{P})}{eCPM_{t,p}(\mathcal{O})} \quad (7.1)$$

If for some order  $o$ , time  $t$  and placement  $p$ ,  $ROI_{t,p}(o)$  is undefined, meaning that  $o$  has not won any impression for that  $time \times placement$  combination, we pick the maximum value of the observed ROIs for order  $o$  as  $ROI_{t,p}(o)$ . This is motivated by the fact that  $o$  did not win impressions for time  $t$  and placement  $p$  in some allocation  $S$ , hence by using the maximum of the observed ROIs for  $o$  the heuristic will not “prefer” this  $t, p$  combination over any other combination. This is what makes the ROI heuristic flexible and able to handle changes in the ad exchange, as it allows the on-line algorithm to pick an unforeseen  $time \times order \times placement$  combination if it is *good enough* for us to disregard the allocation in  $S$ .

## 7.4 The return-on-investment heuristic

The ROIs that we measure from an allocation  $S$  for the budgeted orders are used as a *minimum return on investment requirement* for a budgeted order to win impressions, i.e., if a budgeted order has the highest eCPM of all orders the ROI in that instance must also be greater than the ROI measured from  $S$ .

This is equivalent to adjusting the eCPM of the budgeted orders by

$$eCPM_{t,p}(\mathcal{O}) \leftarrow eCPM_{t,p}(\mathcal{O}) \cdot (1 - ROI_{t,p}(\mathcal{O})) \quad (7.2)$$

This means that our on-line optimization algorithm is identical to the Baseline algorithm in Chapter 4, but with the eCPMs of orders classified as budgeted adjusted according to (7.2).

To reduce our dependence on  $S$ , we can aggregate ROI values. Intuitively, the finer granularity of the ROI measurements, the more dependent we become on the accuracy of  $S$ . We will show that only storing ROIs for each  $placement \times order$  combination renders good results. In this case, aggregating over time, we would pick the most “generous” ROI available.

$$ROI_p(\mathcal{O}) = \min(ROI_{t,p}(\mathcal{O}))$$

As we can see in Table 7.1, ROIs vary very slightly over time.

TABLE 7.1: Measured ROIs per order and placement, over time for LARGE. The average is over 91  $order \times placement$  combinations.

Measure	ROI	Sample Standard Deviation
Average	0.47091	0.00887
Maximum	0.98761	0.04908
Minimum	0.01012	$1.1201 \cdot 10^{-07}$

We can also attempt to aggregate ROIs into  $order \times time$  combinations, but this measure contains greater fluctuations of the ROIs, as seen in Table 7.2. This means having ROIs for each  $order \times time$  combination is a bad idea compared to  $order \times placement$ , and also motivates the need to have ROIs for each  $placement$  instead only having a ROI per order.

## 7.5 Results

We have a set of measurements for each day.

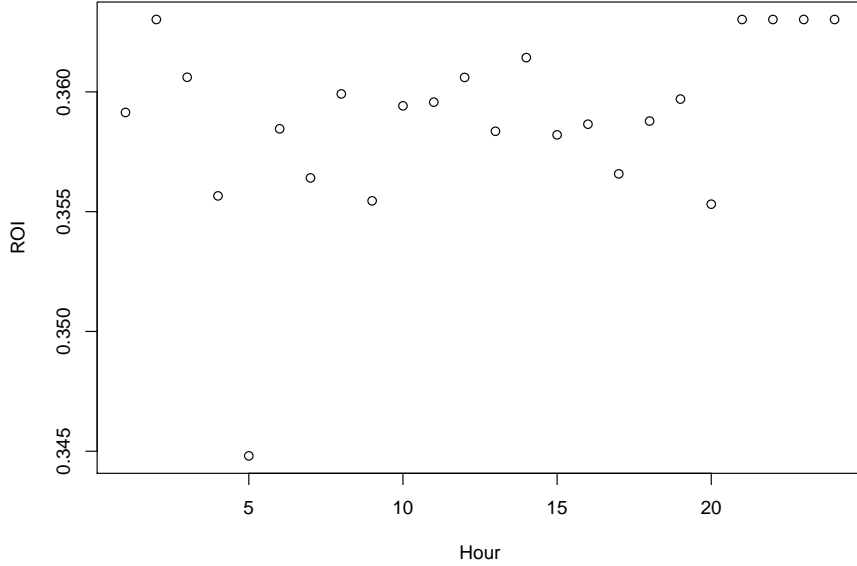


FIGURE 7.2: Example of ROIs measured over time for a placement and order. Notice the slight variations. If we were to use the *order*  $\times$  *placement* ROIs in this case ( $ROI_p(\mathcal{O}) = \min(ROI_{t,p}(\mathcal{O}))$ ) the ROI in this instance would be 0.345.

TABLE 7.2: Measured ROIs per order and time, over placements belonging to LARGE. The average is over 157 data-points.

Measure	ROI	Sample Standard Deviation
Average	0.56368	0.09349
Maximum	0.98760	0.28419
Minimum	0.01011	0.00095

**Baseline:** The Baseline algorithm run with observed traffic as input.

**2D ROI:** Means we have a ROI value for each *order*  $\times$  *placement* combination, with the ROIs derived from the LP solved with predicted traffic.

**3D ROI:** Is the same as 2D ROI but with ROI values for each *order*  $\times$  *placement*  $\times$  *time* combination.

**\*2D ROI & \*3D ROI:** Same as above, but with the ROI values measured from the LP solved with *observed* traffic.

**LP:** A revenue upper bound. The LP with observed traffic as input.

The motivation for having the \*2D ROI and \*3D ROI is that this gives us an indication of how the algorithm would perform if we had *perfect traffic prediction*.

## Revenue

In the following Tables 7.3 through 7.6 we see how different methods of measuring the ROIs perform.

TABLE 7.3: Improvement for the different methods compared to the baseline for MID.

Measure	2D ROI	*2D ROI	3D ROI	*3D ROI	LP
Min	0.570%	4.672%	-0.840%	5.191%	5.341%
Max	4.153%	6.941%	4.066%	7.225%	7.771%
Average	2.503%	5.895%	2.094%	6.203%	6.667%

TABLE 7.4: Improvement for the different methods compared to the baseline for LARGE.

Measure	2D ROI	*2D ROI	3D ROI	*3D ROI	LP
Min	-0.223%	2.829%	-0.890%	2.904%	2.976%
Max	2.762%	6.735%	2.363%	7.954%	8.426%
Average	1.473%	5.331%	0.954%	6.081%	6.547%

TABLE 7.5: Revenue as fraction of optimum for MID (LP solved with observed traffic).

Measure	Baseline	2D ROI	*2D ROI	3D ROI	*3D ROI	LP
Min	92.790%	94.381%	98.581%	93.390%	99.039%	100.000%
Max	94.930%	97.545%	99.615%	97.365%	99.858%	100.000%
Average	93.753%	96.097%	99.276%	95.712%	99.565%	100.000%

TABLE 7.6: Revenue as fraction of optimum for LARGE (LP solved with observed traffic).

Measure	Baseline	2D ROI	*2D ROI	3D ROI	*3D ROI	LP
Min	92.229%	93.245%	98.096%	91.471%	98.860%	100.000%
Max	97.110%	98.055%	99.857%	98.071%	99.929%	100.000%
Average	93.870%	95.248%	98.863%	94.766%	99.564%	100.000%

## Budget utilization improvement

Instead of only looking at the revenue, we can observe *how* we consume the budgets of orders. We do this by measuring how many impressions we use in order to consume the budget.

By measuring how many impressions are assigned to each order using the different algorithms, we get an understanding of why the revenue increases.

TABLE 7.7: The number of impressions assigned to both budgeted and non-budgeted orders for the different ROI methods. The values are relative the Baseline.

Orders	2D ROI	*2D ROI	3D ROI	*3D ROI	LP
MID: Budgeted orders	93.569%	88.157%	92.855%	88.967%	89.012%
MID: Non-budgeted orders	104.543%	106.705%	104.946%	106.064%	105.817%
LARGE: Budgeted orders	91.209%	81.740%	90.129%	77.818%	76.366%
LARGE: Non-budgeted orders	105.347%	109.615%	106.078%	111.393%	112.167%

From the results presented in Table 7.7 we see that for the budgeted orders, all of the ROI methods use a fraction  $< 1$  of the amount of impressions used by the Baseline. Consequently this increases the number of available impressions for the non-budgeted orders, thus increasing the total revenue.

### Budget distribution improvement

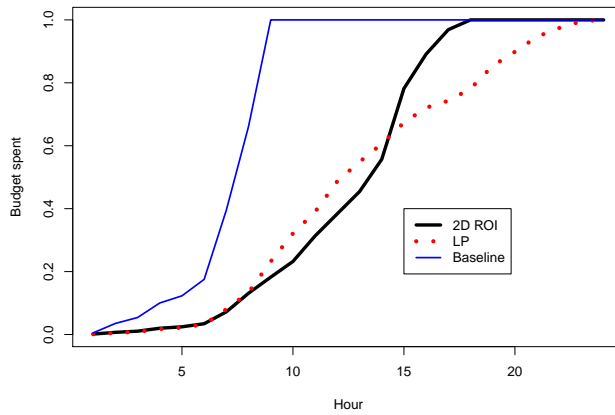


FIGURE 7.3: How the different methods spend the budget of example order #1.

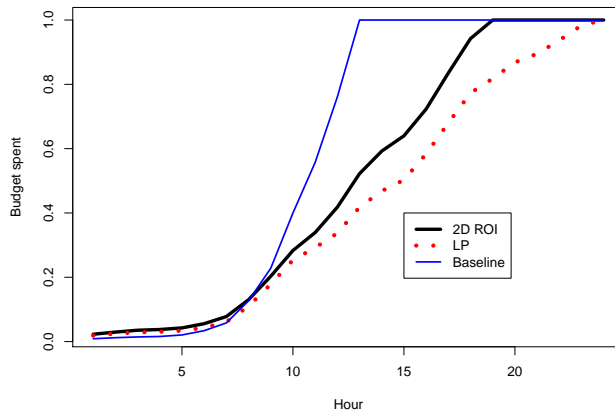


FIGURE 7.4: How the different methods spend the budget of example order #2.

Another property of the ROI heuristic is that it further spreads how the budgets of the orders are being spent. Our Baseline will spend the budgets very quickly at the beginning of each day if allowed. The ROI heuristic slows down the spending, which is a desirable property as advertisers most likely

want their materials shown evenly through the day. The ROI heuristic is not as good as the LP-model, but it is clearly better than the Baseline.

### Summary

As observed in the previous sections, we see that \*2D ROI and \*3D ROI are very close to each other in revenue, with \*3D ROI being slightly better. If we compare 2D ROI and 3D ROI, 2D ROI performs better. This is not a contradiction, but rather implies that, as pointed out earlier, the finer the granularity of the ROI matrix, the more we depend on the allocation  $S$ , which in our case depends on the accuracy of the traffic predictions.

We can also observe, that both \*2D ROI and \*3D ROI are very close to optimal (both  $\geq 98.5\%$  of optimum). This means that the ROI method has the potential to produce very good results, but that the quality of the ROI solution is heavily dependent on  $S$ , again, which in our case depends on the quality of the traffic prediction.



## Conclusion

We have developed and prototyped a budget optimization model that has the potential to increase the revenue of publishers using ad exchanges by several percent.

To achieve this, we have developed a complete budget optimization model.

We created an LP model that solves the problem given off-line data. The LP model proves that there is potential for optimization, and also gives a revenue upper bound for any method in the same context as the model. We can measure theoretical revenue improvements in the region of 6% in our case-study data, but we also show that this figure depends on how many orders consume the total of their budget.

This measure of 6% is relevant since it is the LP model compared to a greedy Baseline algorithm that closely resembles the production algorithm, adapted to the simplified domain we consider in this thesis.

To make the LP useful for solving the on-line part of the problem we developed a linear regression model for predicting placement impressions. We have shown that the traffic prediction model can render up to 20% better accuracy compared to using observed traffic for the previous day as a prediction.

If a placement has insufficient training-data to be predicted, we can use the rudimentary method of copying yesterdays traffic for that placement as a prediction.

By combining the LP with the traffic prediction we can get a predicted optimal  $placement \times material \times time$  allocation  $S$ . We classify orders as budgeted and non-budgeted depending on if an order consumes its budget in  $S$  or not. From  $S$  we show that it is possible to measure differences in eCPM between budgeted orders winning impressions and non-budgeted orders not winning impressions. These measurements can be used as a minimum *return on investment* requirement for budgeted orders in the on-line scenario, making these orders consume their budgets using less impressions than in the Baseline.

The return on investment heuristic renders higher revenue compared to the Baseline, and helps the orders spend their budgets more evenly throughout the day and on the placements that produce the highest return on investment. With perfect traffic prediction, an optimal allocation  $S$  when used with the ROI heuristic can reproduce up to 99% of the LP revenue for  $S$ . With imperfect traffic prediction, an allocation  $S'$  used with the ROI heuristic renders about 95% of the revenue of the LP output solved for  $S$ .

The reason the revenue increases is because the budgeted orders consume their budgets using less impressions, allowing more impressions for non-budgeted orders, rendering an over-all increase in revenue for the publisher.

## 8.1 Further work

In Chapter 7 we could see that if the traffic prediction is perfect, the return on investment method has the potential to render almost optimal revenue. But when we derive ROIs from our prediction model we are not able to achieve the optimal revenue.

One step to improve the revenue is to develop a more sophisticated traffic prediction model. If we can make closer to perfect traffic prediction, the revenue realized by the ROI heuristic will go up.

### Control theory for adjusting ROIs

Another potential change is to implement methods of Control Theory to adjust the ROIs on-line. Chen et al. in [CBAD11] do exactly this and claims that they are able to render a revenue improvement. A motivation of having adjustments of the ROI on-line can be seen in Figure 7.2. The ROIs fluctuate ever so slightly over time. In our case we pick the minimum of the measured ROI values for use in the on-line algorithm. We saw in Tables 7.3 & 7.4 that if our traffic prediction is perfect, having ROIs for each *placement*  $\times$  *order*  $\times$  *time* increased revenue the most. If our traffic prediction was imperfect, having ROIs for each *placement*  $\times$  *order* increased revenue the most, since these ROIs were less dependent on accurate predictions. It may be the case that by having on-line adjustment of the ROIs it is possible to find some sort of middle-ground between the two cases.

### Running the budget optimization model several times a day

As an alternative to Control Theory for adjusting the ROIs we could potentially modify the LP and traffic prediction model to be able to run several times a day. For example: at the start of each day we could predict the traffic for all placements for the whole day, solve the LP and measure the ROIs to adjust the eCPMs.

After some time has passed (e.g. a couple of hours), we predict the traffic for the remaining hours of the day, but this time with a more accurate traffic prediction model, since we saw in Chapter 6 that it is possible to predict hours more accurately than days. Solving the LP for the remaining hours with the new prediction we would measure new ROIs and then re-adjust the eCPMs in order to take the most recent available information into consideration. This could potentially render a higher increase in revenue compared to only doing it at the start of each day; and since the eCPMs are adjusted more frequently the method is more flexible regarding unforeseen changes in both web traffic and the ad exchange.

## Bibliography

- [BCI<sup>+</sup>07] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Kamal Jain, Omid Etesami, and Mohammad Mahdian. Dynamics of bid optimization in online advertisement auctions. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, pages 531–540, New York, NY, USA, 2007. ACM.
- [CBAD11] Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R. Devanur. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11*, pages 1307–1315, New York, NY, USA, 2011. ACM.
- [Chv83] Vašek Chvátal. *Linear programming*. W.H. Freeman, 1983.
- [FHK<sup>+</sup>10] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In Mark de Berg and Ulrich Meyer, editors, *Algorithms – ESA 2010*, volume 6346 of *Lecture Notes in Computer Science*, pages 182–194. Springer Berlin / Heidelberg, 2010.
- [FIMN08] Ureil Feige, Nicole Immorlica, Vahab Mirrokni, and Hamid Nazerzadeh. A combinatorial allocation mechanism with penalties for banner advertising. In *Proceedings of the 17th international conference on World Wide Web, WWW '08*, pages 169–178, New York, NY, USA, 2008. ACM.
- [MA02] J. Susan Milton and Jesse C. Arnold. *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*. McGraw-Hill, Inc., New York, NY, USA, 4th edition, 2002.
- [MNS07] Mohammad Mahdian, Hamid Nazerzadeh, and Amin Saberi. Allocating online advertisement space with unreliable estimates. In *Proceedings of the 8th ACM conference on Electronic commerce, EC '07*, pages 288–294, New York, NY, USA, 2007. ACM.
- [ZCL08] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. Budget constrained bidding in keyword auctions and online knapsack problems. In Christos Papadimitriou and Shuzhong Zhang, editors, *Internet and Network Economics*, volume 5385 of *Lecture Notes in Computer Science*, pages 566–576. Springer Berlin / Heidelberg, 2008.

## Glossary

**Ad exchange** A system facilitating buying of selling on-line media advertising inventory from multiple networks (orders).<sup>1</sup>

**Admeta** The company where this master's thesis was written.

**Advertiser** A company that places orders to publishers.

**Budget** Budgets are assigned to orders for the duration of the advertising campaign (total budget). Optionally, orders can have a daily budget.

**CPC** Cost-Per-Click. Specifies the amount the advertiser pays if a visitor clicks the advertisement.

**CPM** Cost-Per-Milli. Specifies the amount the advertiser pays for every thousand views.

**eCPM** estimated Cost-Per-Milli. The combination of a click-through-rate and advertiser bids. Estimated revenue for showing a material one thousand times.

**CTR** Click-through-rate. The amount of clicks a placement receives per thousand views.

**Impression** An impression is when a visitor views a placement on a site.

**Material** A material is the actual advertisement. Materials are shown on placements. There are three types of materials.

**Creative** Images or Adobe Flash advertisements.

**Text** Advertisements containing plain-text. Placements that accept text materials have a number  $n$  associated with them, expressing how many text ads fit within the placement.

**Order** Orders are placed by advertisers, and sent to publishers. Orders have budgets, one or more materials, an optional end-date and a specification which of CPM and CPC the advertiser is willing to pay for.

**Placement** A placement is a “physical” area with on the publishers site where materials can be placed.

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Ad\\_exchange](http://en.wikipedia.org/wiki/Ad_exchange)

---

**Publisher** A publisher is the user of the Tango system. Publishers have one or more sites where they own placements, and they want to fill this placements with materials subject to maximizing their revenue.

**ROI** Return on investment. eCPM differences that can be measured from a *placement × order × time* impression allocation.

**Revenue** The revenue of publishers is in direct proportion to CPM and CPC.

**Visitor** A person visiting a website.