Deriving a Marketing Budget Allocation Model Under Uncertain History Data

by
Michel van Gelder (438662)
B.Sc. Tilburg University (2014)
Supervised by: Dr. J.C. Vera Lizcano

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Operations Research and Management Science Tilburg

School of Economics and Management
Tilburg University

In association with:

June 23, 2016

Contents

Management Summary

This thesis considers a media budget allocation model over AdWords campaigns, as is encountered in real-life by the company PauwR and probably by many others. The underlying budget allocation model is kept relatively simple, and only considers budget constraints and lower and upper bounds on investments per (group of) campaigns. The hard part, and also the main topic of this thesis, is to model the campaign-specific sales functions, whose sum is the objective function of the model considered.

The campaign-specific sales function depicts the relation between invested budget in a campaign and the expected resulting revenue from that campaign. From literature we know that a sales function should be a monotonically non-decreasing concave function. To perform a regression with this particular shape we propose a Convex Hull Method based upon Aguilera et al. (2010), as well as a more direct Second-Order Cone formulation of the problem. These algorithms both result in deterministic estimations of the sales functions.

However, the main problem considered is to account for data uncertainty when estimating these sales functions. This uncertainty follows from the lack of information with multiple considered data sources. This uncertainty is accounted for by introducing a robust variant of the Second-Order Cone formulation, to deal with uncertainty in both cost and revenue. Moreover, due to flawed data or unrelated occasional events, outliers occur in the data. Hence, in addition to the previous algorithms, an outlier exclusion is performed using a Simulated Annealing algorithm.

Finally, the algorithms are tested on artificially generated data sets, as well as on real data from a client of PauwR. An R-squared score is used to test the performance of the different proposed algorithms. From these scores we conclude that the algorithms perform properly on the artificial data, in sharp contrast to the poor performance on the real data sets. We also conclude that the Convex Hull Method is inferior to the Second-Order Cone formulations. The last substantial conclusion that is drawn, is that the necessity for a robust model is questionable and whether it would be possible to obtain reasonable estimates for the sales function with the current level of uncertainty.

In conclusion, the models could be implemented when the desired concave shape is evident in the data and the level of uncertainty is within bounds. More specifically, when the R-Squared value of the regressions is more than a desired lower bound even with the uncertainty considered. In the current setting of the client of PauwR this is not the case and the algorithm is too inconclusive to be implemented directly. Hence, more research is needed.

1 Introduction

According to the American Marketing Association, Marketing is the activity, set of institutions, and processes for creating, communicating, delivering, and exchanging offerings that have value for customers, clients, partners, and society at large. Based upon this definition, the most renown form of marketing is the exposure of a targeted audience to advertisements distributed via different media sources. These media sources can be split-up into multiple categories, with online marketing being the category of interest in this thesis.

In present-day, the range of marketing channels has rapidly developed through the introduction and swift expansion of internet in the past 20 years. Digital marketing already accounts for a considerable portion, often 30-70%, of the marketing mix, and is expected to increase even further in the upcoming 5 years according to CMO Council (15 Dec. 2015). Due to Google Inc. being the most widely used search engine and the company behind the paid link-service AdWords, a vast amount of the media used for internet marketing is controlled by Google Inc. In addition, Google also keeps track of all allowed, and interesting, statistics regarding marketing campaigns and their results. These statistics are presented in Google Analytics and can be retrieved from there as well. Usually, the only involved parties who are authorized to view these statistics are the company who offers the products and an external marketing agency. The latter, at least partially, owing her existence to the complexity and seemingly endless options present in current marketing structures. The complexity and ample variety of options of this issue, combined with the available data, lead to an interesting optimization problem in general.

Marketing is widely accepted as being of great importance, if not indispensable, for a company to be profitable. This causes companies nowadays to allocate substantial yearly budgets to online marketing. Since these companies do not want the relatively big budgets to be spend in vain, the question arises whether there exists an optimal distribution of these budgets among their available AdWords campaigns. Hence, the modeling of this budget allocation and the possible derivation of a mathematically optimized distribution of a marketing budget could benefit companies greatly.

PauwR is an online marketing and media agency, located in Tilburg. PauwR works with many clients across the Netherlands, and is therefore responsible for the allocation of many company's (online) marketing budget. PauwR is a progressive organization, striving to stay ahead of the competition in online marketing. It is also one of the few Dutch organizations who incorporates the See-Think-Do model. This model states that nearly all of the marketing campaigns can be roughly split-up into three phases. The See-Phase inhibits the creation of brand-awareness, where the Think-Phase contains the consideration and evaluation of the brand by the customer, followed by the actual conversion phase; the Do-Phase. These phases each inherit their own type of AdWords campaigns.

In addition, with all of these phases PauwR works closely together with Google Inc. and their services. This enables them to accurately measure all of the relevant variables and to adjust their campaign parameters wherever they see needed, and whenever they see needed. PauwR believes this collaboration is going well, but thinks they are not yet able to make most out of all the services offered by Google Inc. To stay ahead of the competition, they would like to make some steps into the transition towards a more data-driven approach. This introduces the main topic of this thesis; to develop a data-driven approach to distribute the given budget over the possible AdWords campaigns, based upon the available history data.

The remainder of this Master Thesis is structured as follows. In Chapter 2 a problem description is given from a practical point of view, with a listing of relevant factors that are included in the problem at hand. Chapter 3 discusses compatible literature on the subject. Although the exact problem is not yet solved in current literature, comparable problems and relevant solution methods are of importance for the solution approach presented in this thesis. Chapter 4 provides a theoretical background on the subject, as well as a mathematical model that follows from the relevant factors presented in Chapter 2 and the reviewed literature from Chapter 3. Chapter 5 is the main section of this thesis and includes the solution approach to the model presented in Chapter 4, with additional improvement steps and the robust counterparts for the regressions needed. Chapter 6 provides the implementation of the methods derived in Chapter 5 into a practical case with real data from a client of PauwR as well as artificially generated data sets. Consecutively, in Chapter 7 the results are presented and discussed. This thesis is concluded with a conclusion and discussion in Chapter 8.

2 Problem Description

The main goal of a marketer is to divide a given budget among different possible media channels, in such a way that the given objective is maximized. In the majority of the cases, the given objective is to generate the largest possible revenue, which is also assumed to be the intention in this thesis. There are other possible objectives that could be considered, e.g. the number of views of a website, the number of transactions following a marketing campaign or exposure in general.

2.1 Constraints and Complexity

The marketer's division of the available budget is subject to multiple constraints. The most prominent constraint treated in this thesis is the information uncertainty. Most information regarding the marketing campaigns, their expenditure and revenues can be retrieved from Google Analytics. However, for the expenditure parameters described in the next paragraphs, uncertainty is present due to incomplete or averaged information. In addition to this uncertainty, there also exist multiple fluctuations due to untraceable events. This, again, causes a significant increase in the complexity of the problem.

However, even if the uncertainty would not be an issue, multiple constraints would be imposed on the marketer's problem leading to a non-trivial problem. Multiple factors are complicating the situation at hand, of which the most important are:

Budget limitation

The first, most obvious constraint, is a finite budget. This finite budget limits the marketeer, preventing him/her from using every marketing channel to its full potential. This implies that concessions have to be made regarding the usage of channels, and a desirability of an optimal allocation under the budget constraint. Evidently, the impact of the budget limitation on the problem statement varies between companies and over time. During times of recession, when marketing budgets are especially vulnerable to cutbacks, these budget limitations have proven to be very tight in many cases.

Expenditure parameters

In addition to varying budgets, there are also differing expenditure parameters among the given channels. Each channel provided gives different results regarding different assigned budgets, and a varying rate of return when assigned different budgets. More specifically, there is a diminishing rate of return in each marketing channel. This is also the property that excludes a single optimal channel in which one should invest the total budget, regardless of the available budget.

Inter-Channel competition

The third factor to be taken into account are the dependencies of different marketing channels. More specific, the overlap in the audiences reached with these channels. For example, if one invests in television ads and in radio commercials, there will generally be a certain part of the audience that is reached with both of these instruments. This entails the inability to simply add the expected results of the two channels in our problem setting. The relations between different channels and the overlap between the audiences needs to be taken into account. This could not only be the case for completely distinct media channels, but also for the diverse types of online marketing channels. Undoubtedly, the magnitude of these overlaps among mutual pairs of channels differs.

Auctioneer's and final bid-placer's influence

The last major component adding complexity considered here, is the influence of the auctioneer and final bid-placer, Google Inc., in the evolution of a daily budget into a certain revenue generation. Since Google Inc. determines the bids placed on different ads for different keywords on different times of the day, for both the company at hand and most of his competition, Google has a big impact on the costs accompanying the generated profits. Even in this perfect-information world, the influence of Google is unmistakably a factor that needs to be considered. Combining this influence with the competition of a company's own set of marketing campaigns already leads to a major increase in complexity of the problem.

2.2 PauwR-Specific Problem

Although the problem addressed in this Master Thesis has been studied in the literature, as will be seen in Section 3, this exact problem is yet to be solved. The cause for this lies in the diversity of the known marketing problems, with corresponding rapid innovations in the marketing business, but foremost in the previously described complexity of the problem. Due to these properties, there is no exact solution for this problem in general, leaving two options; to tackle a specific case of the general problem, or to reduce the exactness of the solution. In this thesis a combination of both methods shall be implemented to create a solvable problem statement while staying as close to reality as possible.

The diversity mentioned previously already implies choices to be made regarding the problem to be solved. This gives rise to a more specific case to be considered. Moreover, the rapid innovations indicate a continuously changing set of options and possible constraints, leading to the necessity of the usage of the most modern available tools, in order to solve a, currently, relevant problem. Since PauwR is a Internet Marketing company, the scope of this thesis is already implicitly limited to online marketing. However, due to reasons stated, the scope is narrowed even further.

The assigned marketing budget towards online marketing by clients of PauwR, is split-up further into the following categories:

Social Media

This includes advertisements on social media websites and applications, such as Facebook, Instagram, LinkedIn, etc. The main focus with PauwR is on social media which is trending in the Netherlands. There are exceptions where smaller campaigns are run for, e.g. Yelp.

Organic Searches (including Search Engine Optimization(SEO))
This includes potential customers en visitors generated by searches of individuals via search engines such as Google and Duck-Duck-Go. This category also includes SEO, which optimizes the probability of a user finding a clients website or direct link via a search engine.

Direct Traffic
This includes customers directly typing in the URL to a website, or a specific page on a website.

Paid Searches (AdWords)
This includes visitors generated via sponsored links with popular search engines.

With the AdWords service of Google Inc. marketers are enabled to set up online marketing campaigns for these paid-search links. Marketers are able to construct campaigns, mainly consisting of keywords and assigned daily budgets. Apart from these main parameters there is an abundance of optional specifications, e.g. time-preferences and maximum costs-per-click, that can be specified. However, in general these are left at their default values by PauwR and are assumed to have these values as well in this thesis.

So, when the campaigns are setup with their keywords and the budgets are allocated, which is the main topic of this thesis, Google basically distinguishes the following steps, according to Google Support (June, 2016). Whenever an individual types in one or more from the specified keywords presented in a campaign, a bidding occurs to determine which paid search links will be shown to the potential customer. For these links, Google has predetermined 3-7 reserved spaces for the top 3-7 links to be shown. The top links are elected via a predetermined formula from Google Inc., which depends on your bid and a predetermined quality score. Your bid is the amount you promise to pay Google Inc., i.e. how much is subtracted from your assigned budget, when the potential customer clicks your appeared link. Let it be emphasized that you do not pay anything when your link is not clicked. The quality score on the other hand, represents how well you fit the customer needs according to Google Inc. This score follows from user experience with your website along with the keywords typed in, your URL and your specific ad, making up your relevance.

Please note that all factors in play could be controlled by the companies themselves, but in general are regulated by Google Inc. This principle is justified by the motive of Google Inc.'s need to match the customers to the companies as well as possible. This prevents the system from collapsing upon itself due to antagonistic motives of the monopolistic influence contender.

From these four categories, Paid Searches and Social Media are the two most interesting categories, due to the control that can be exercised over these categories and the easily measurable and collectible data. Apart from SEO, Organic searches and Direct traffic cannot be optimized in the same manner, due to them being 100 percent directly dependent on user input.The issue with SEO however, is the large number of factors that impact the results of the budget spend on this aspect. Hence, SEO cannot be forecasted accurately, at least by PauwR, and is left out of the scope of this thesis.

Hence, this thesis shall mainly focus on Paid searches, due to the ability to measure results and the amount of controllable variables. Paid search also has a significant correlation to the other three categories stated before. A significant number of customers (about 30% for the PauwR client considered in this thesis) do not purchase immediately upon finding the desired product via such a paid search link. A significant part of the purchases follows from returning customers, who have taken a few days to think about their possible investment. These customers could be returning via the same campaign, an organic search or direct traffic, even though their original click-bait was one of the paid search links. Evidently, it is also possible for customers to return via a paid search link corresponding to another AdWords campaign, yielding inter-dependencies in the campaigns.

As stated before, Google Inc.'s policy is a huge factor regarding the distribution of the budgets presented for paid search links. When a budget is allocated to a Paid Aearch campaign, Google Inc. distributes this budget over the day. Since Google Inc. is also in control of the bidding process for most other companies, Google Inc.'s policy has a very large impact on the number of views, prices et cetera. Since PauwR does not have any influence in this process, and does not wish to have any influence on the bidding process, this is assumed to be exogenous. This leads to the objective of the problem remaining to be just the division of the budget among the different possible AdWords campaigns, also known as campaigns. However, the budget distribution by Google Inc., and the fact that you only have to pay when someone clicks on your link, do lead to a significant level of uncertainty, and hence a harder optimization problem.

The further concessions made on the exactness of the solution due to problem complexity arguments will be further discussed in the theoretical background and solution approach sections.

2.3 Thesis Objective

The main objective of this master thesis project is:

To develop a mathematical methodology which optimally estimates the generated revenue as a function of investment, subject to uncertain history data, which is to be given as the objective function in a budget allocation model

3 Literature Review

In this thesis a mathematical model is proposed for budget allocation of AdWords budgets. To construct this model, we have considered literature regarding marketing theory and existing models solving similar problems. From this literature the need arose to review literature on convex regression models, outlier exclusion and robust optimization. The studies inspected for this thesis are stated in the following sections.

3.1 Marketing Theory

As input for the model that is constructed in this thesis, we are given the daily costs and revenues for the marketing channels. Intuitively the revenues can be described as an increasing function in costs, with decreasing marginals. Bhattacharya (2009) states that this relation is S-shaped as in Figure 1, and that it can be described by a Gompertz function for example. This shape is also supported by Giagkoulas (2011). However, this S-shape follows from consequential factored setup costs of marketing channels. With AdWords campaigns, these setup costs are negligible, and the monotonically non-decreasing concave relation of the right hand side of the S-shape remains, Adwords Support (2016).

In Bhattacharya (2008) challenges faced when solving budget allocation models are described. Some of the most important challenges are a lack of good data, interaction effects of marketing channels, dynamic effects such as advertising leads, multicollinearity and non-linear functional forms. The interaction effects are also of great influence on the model that Giagkoulas (2011) presents on his particular case of a budget allocation model. However, multicollinearity need not considered when we limit the scope to AdWords campaigns, because we only consider one possible medium of advertising. Further, with the available data of Google Analytics, most of the lack of good data is no longer an issue. In Chapters 5 and 6 the data that cannot be obtained from Google Analytics in our situation is accounted for by introducing the concept of data uncertainty. Moreover, the interaction effect and dynamic effect challenges can be estimated via the data available in Google Analytics, as is further explained in Section 6.1. This implies that these factors can be estimated and thus remove this issue, but lead to a new challenge of data uncertainty. Lastly, the non-linear functional form is still a challenge in this thesis, but is easier since we do not consider setup costs.

3.2 Budget Allocation Model

Giagkoulas (2011) solves a similar budget allocation model as the one in this thesis. He stated a mathematical model which maximizes the revenue function subject to a budget constraint and a lower and upper bound on the budget spend per channel. These constraints also hold true for the model in this thesis.

Bhattachary (2010) and Bhattacharya (2012) also proposed a solution method for a budget allocation model. However, this Generalized Additive Model for

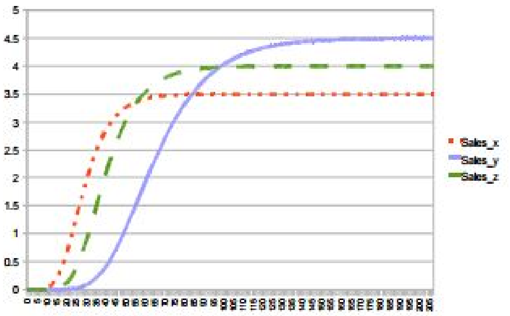

Figure 1: The example function of the relation between sales and costs of a marketing channel presented in Bhattacharya (2009)

Marketing Mix Modeling was simulation based, and is mainly useful for the determination of time-lags in marketing. This can be, as stated before, retrieved from the Google Analytics data for this thesis. Moreover, Bhattacharya (2010) already states that the method is very simplistic and should mainly be used to determine which time-lags are significant.

Lastly, in Naik et al. (2005) an extension to the already existing Lancaster model is proposed. This paper is mainly focused on how different competing marketeers interact. A dual methodology is proposed in this paper, the first model is calculating the optimal combination and the second model estimates the resulting effect. The paper basically used the Lancaster model (J. Lancaster, 1966) and extends it by introducing interaction effects and multiple brands. This problem formulation is not necessarily applicable in this thesis due to a different focus. However, the paper does provide a good insight in the possible interaction effects that need to be taken into consideration.

3.3 Concave Regression

Following Giagkoulas (2011), Bhattacharya (2010) and Bhattacharya (2012), the objective function of the mathematical model should consist of at least one monotonically non-decreasing concave function. To determine this function, a regression needs to be done. One method to perform this regression, explored in literature, is the Convex Hull Method by Aguilera et al.(2010). This method uses a LOESS regression method, presented in Cleveland (1979) to get a smooth estimator. This LOESS regression of Cleveland (1979) is an extension to the local fitting of polynomials as presented in Macauley (1931) and uses parameters based on Lutenegger (1979). The algorithm of Aguilera et al. (2010) then takes a convex hull of a subset of this smooth estimator to get a convex regression function. Jacoby (2000) provides a more recent discussion on the LOESS regression method as well as the parameter choices.

The second method discussed in this thesis is a direct Quadratic Formulation of the regression problem. This Quadratic Formulation is based on the properties and implementations described in Lobo et al. (1998) and is supported by the findings of Miyashiro and Takano (2013). It is then implemented as proposed in Alizadeh and Goldfarb (2004) and Boyd and Vandenberghe (2004). As a solver for this SOCP formulation, MOSEK is chosen, based on the characteristics stated in Anderson (2013).

3.4 Outlier Exclusion

Many real-world data sets have to deal with outliers. Their possible negative effects and power are discussed in J.w. Osborne & A. Overbay (2004). Since this applies to the problem at hand as well, and the effects could undermine the outcomes of the regressions performed, outlier exclusion is performed. This is done via a Simulated Annealing algorithm, based upon the algorithm presented in Corana et al.(1987) and Dell'Amico et al.(2009), using the Metropolis Criterion first introduced in Metropolis (1953).

3.5 Robust Optimization

The data in this thesis contains uncertainty, which has to be accounted for. There exists extensive literature on robust optimization, which deals with data uncertainty. Important papers are the general papers about robust optimization of Sim (2004) and B.L. Gorissen et al.(2015). A more in-depth explanation regarding data-driven optimization is given in Bertsimas et al. (2013) , where Goh & Sim (2010) discuss tractable approximations of distributionally robust optimization problems. Furthermore, in this thesis the method presented in Bertsimas & Thiele (2006) is followed to formulate the robust counterpart. This is then solved with the Adversarial Approach, first described in Bienstock & Ozbay (2008).

4 Theoretical Background

In this chapter, a detailed description of involved factors and terminology is presented, to better understand the problem and the modeling hereof. From this theoretical knowledge, logical steps follow, which lead up to the definitions of the problem's decision variables, parameters, sets and objective function. Which will be brought together in the mathematical problem formulation, with its constraints and objective function as well as all its sets, parameters and variables.

4.1 Definitions

To fully understand the remainder of this thesis, we introduce the terms that will be used extensively throughout, with their respective definitions. These are presented below:

Marketing Channel
: According to Black's Law Dictionary (2016) "a Marketing channel is the path from vendor to the consumer of a company's goods and services, flowing in one direction". In this thesis it is used as a general term referring to an available media to broadcast the client's message.

AdWords Campaign
: Google Adwords Support (2016) states that "an AdWords campaign, or campaign for short, is a set of ad groups (ads, keywords, and bids) that share a budget, location targeting, and other settings. Campaigns are often used to organize categories of products or services that you offer".

Marketing Budget
: The Online Business Dictionary (2016) states that a "Marketing budget, or budget, is an estimated projection of costs required to promote a business' products or services. A marketing budget will typically include all promotional costs, including marketing communications such as website development, advertising and public relations, as well as the costs of employing marketing staff and utilizing office space". Here it will be referred to as the amount of money that is allocated to marketing, a specific campaign or a set of campaigns.

4.2 Problem Modeling

The optimal budget allocation over the possible campaigns implies that the available amount of money should be invested in a subset of the campaigns and diverted among this subset in such a way that the total revenue generated by the total set of campaigns would be maximized. This thesis starts out by modeling the general version of this problem, assuming independence of the campaigns and their respective revenues. This independence, as can seen in Chapter 6, will be a weak assumption when implementing some simple transformations on the input data.

The first computation of the mathematical model of our problem presented is deterministic of nature, implying that uncertainty is neglected. This follows directly from the deterministic way in which the objective function is computed. Namely, for the deterministic model, for a given set of campaigns C, we define the total sales (revenue) function generated by a given budget allocation as

$$S(b) = \sum_{c} s_c(b_c) \tag{1}$$

where b is the vector of budget allocated and s_c the corresponding sales generated by the allocation of b_c to channel c.

As stated in Chapter 3, the literature suggests non-negative, yet diminishing, marginal returns on investment per channel. Hence, this implies a monotonically non-decreasing concave shape for $s_c(b_c)$ for all channels. We may conclude that S(b) is a monotonically non-decreasing concave function, as it is a sum of the monotonically non-decreasing concave functions $s_c(b_c)$. The theorem to support this claim, as well as the definitions of convexity and concavity, are given in Appendix B.

Please note that in the solution approach in Chapter 5, the deterministic version of the problem will be solved initially, followed by several steps leading towards the solution of the more realistic Robust Counterpart. This can be readily done by replacing the deterministic regressions to estimate S(b) by their robust equivalent, as is seen in Sections 5.2 and 5.3.

4.3 Mathematical Problem Formulation

The final step considered in this thesis is to solve a model which optimally allocates our budget, using the estimated revenue function as its objective value. In doing so we want to maximize the revenue under a budget constraint. Also, there are no short positions, i.e. all investments need to be positive. This implies that the deterministic version of our problem needs to be solved by the following model:

Budget Allocation Model

Input: A maximum budget (B) and a set of possible campaigns (C) with their history data sets H^c, which are explained into more detail at the end of this section and are used to construct the objective function

Model:

Functions:

- $S : \mathbb{R}^C \to \mathbb{R}$: (Total) Sales function, assumed to be equal to the sum of individual sales functions per campaign, i.e. $S(b) = \sum_c s_c(b_c)$

Parameters:

- $B \in \mathbb{R}_{++}$: Maximum Budget to be spend
- $LB_c \in \mathbb{R}_+$: Lower Bound for budget allocated to campaign c
- $UB_c \in \mathbb{R}_+$: Upper Bound for budget allocated to campaign c

Decision Variables:

- $b_c \in \mathbb{R}_+^C$: Budget allocated to campaign c

Mathematical Problem:

$$\text{(P)} \quad \max \sum_c s_c(b_c) \tag{2}$$

$$\text{s.t.} \sum_c b_c \leq \text{B} \tag{3}$$

$$b_c \geq LB_c \quad \forall c \in C \tag{4}$$

$$b_c \leq UB_c \quad \forall c \in C \tag{5}$$

Constraint (3) denotes the budget limitation which was introduced in Chapter 2.1, ensuring that the sum of the invested budget per channel does not exceed the total available budget. In addition to this constraint, equation (4) imposes a lower bound on the amount invested per channel. In this model, the lower bound can be set for user preference. Otherwise it is assumed to be 0, such that it prevents an, unrealistic, short position in any campaign. Equation (5) imposes an upper bound for the expenditure on every single campaign. Similar to the lower bound, this is subject to user input. The default setting of this upper bound is equal to the maximum observed expenditure in the set of history data to prevent undesired extrapolation.

In addition to the standard upper and lower bound constraints given by (4) and (5), one may impose constraints of the following type:

$$\sum_{c' \in C'} b_{c'} \geq LB_{C'} \quad \textit{where } \; C' \subseteq C \tag{6}$$

$$\sum_{c' \in C'} b_{c'} \leq UB_{C'} \quad \textit{where a } \; C' \subseteq C \tag{7}$$

Hence, imposing upper and lower bounds on investment on combinations of campaigns. These are linear constraints, and therefore do not drastically increase the complexity of the model. Constraints of this type are omitted in this thesis, due to the grouping that will be seen in Section 6.1. This grouping ensures that the remaining campaigns can be regarded as mutually independent as far as practical implications go. Thus, it would not be reasonable to constraint the investment on certain combinations of campaigns.

In (P), the objective function in (2) is a maximization over the revenue function (1). Which is a monotonically non-decreasing concave objective function, with linear constraints (3) - (5). Hence, this problem depicts a convex optimization problem.

The only part from the problem formulation left to specify is $s_c(b_c)$. The difficulty however, lies in the derivation of a sensible definition of the channel sales function. This derivation will happen upon the daily history data sets $H^c \in \mathbb{R}^{2\times h}$, with $h = |K|$ and entries $H_k^c = (x_k^c, y_k^c)$ for each day $k \in K$, with $K = 1, 2, ..., h$. The data sets H^c are subject to interaction effects between different marketing channels, varying time-delays, uncertainty in the expenditure data x, uncertainty in the revenue data y and random events causing outliers.The main challenge will be to account for these factors when obtaining the sales function.

Please note that in the upcoming sections the superscript c will be leftout. So, H will be used to denote H^c. This is due to the seperate evaluation of the sets H^c, due to their assumed independence. The same holds for the data points $H_k^c = (x_k^c, y_k^c)$ being denoted as $H_k = (x_k, y_k)$.

5 Sales Function Estimation

In this chapter the main part of the solution approach for the model presented in Chapter 4.3 is discussed; the modeling of the sales function. Section 5.1 starts with the modeling of the sales function in a deterministic approach. To derive the monotonically non-decreasing concave sales functions per campaign ($s_c(b_c)$), two methods are proposed. The first method is the Convex Hull Method and the second method is a more direct Quadratic Formulation of the problem.

Consecutively, Chapters 5.2 and 5.3 then discuss the uncertainties impacting the data sets, and how to cope with this uncertainty. This is done via robust optimization. The possible factors of uncertainty are quantified, and the robust counterparts for the nominal solution of 5.1 are formulated and solved.

In Figure 2, an overview is given of how these algorithms blend together to solve the main problem at hand.

5.1 Nominal Sales Function

Since the sales function is the sum over the expected revenue curves per campaign, the establishment of an sales function per campaign is necessary.

5.1.1 Non-Parametric Regression

To construct the campaign-specific sales function $s_c(b_c)$, a regression on the collected data points will be done. Since we do not expect the shape of the function to be strictly linear, polynomial and/or logarithmic, we start with a non-parametric regression model. We define the dependence of every entry y_k on x_k as

$$y_k = f_c(x_k) + \epsilon_k \quad \forall k \in K$$

with $y_k \in \mathbb{R}_+$, $x_k \in \mathbb{R}_+$ and $\epsilon_k \in \mathbb{R}$ as an unknown error term. As stated in Chapter 4.2, $f_c(\cdot)$ is assumed to be a monotonically increasing concave function. These known characteristics give rise to multiple possible approaches to derive a well fitted function $f_c(\cdot)$. The first approach considered is a convex hull method proposed by Aguilera et al. (2010). The second approach acknowledged is a Second-Order Cone Problem (SOCP) formulation which directly states and implements the non-decreasing concavity constraints and maximized fitting objective.

As we want to estimate the sales function f_c for a fixed channel c, we drop the subscript c and we will call the function f.

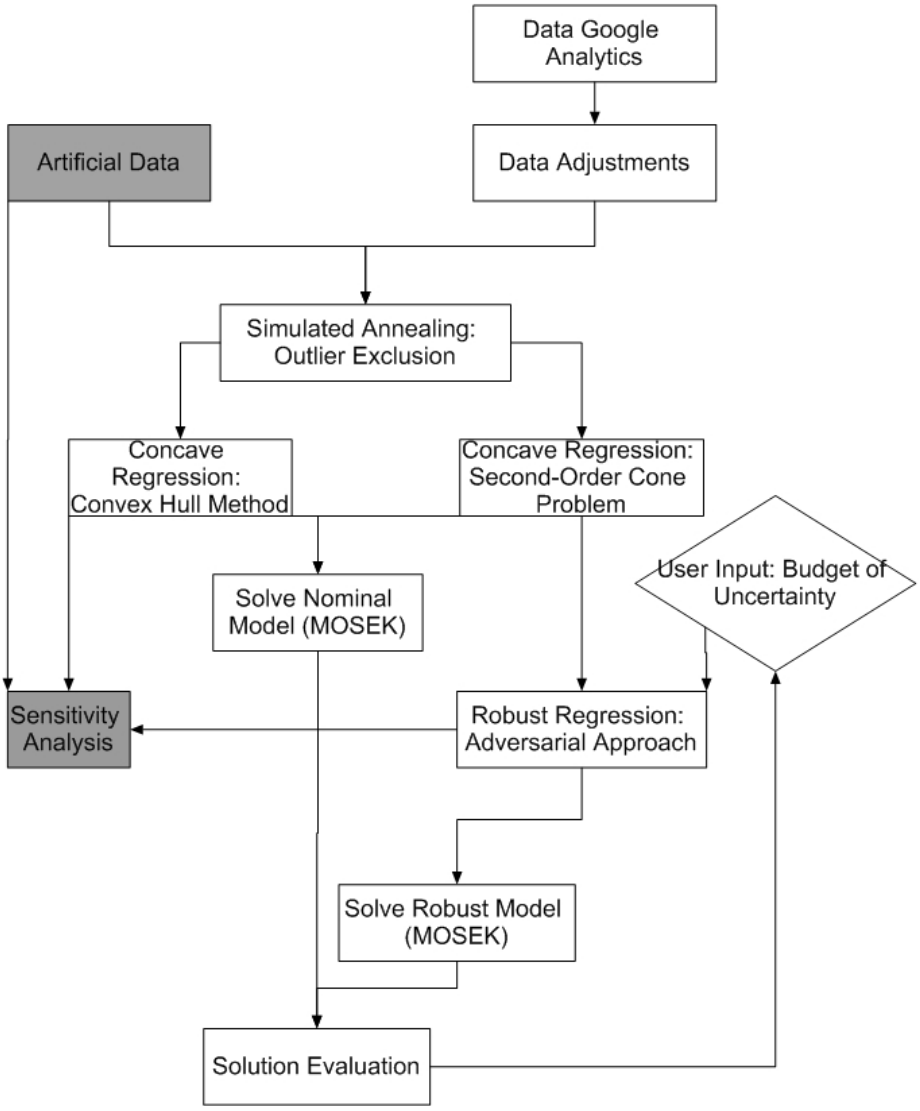

Figure 2: FlowChart of the solution approach. Gray boxes represent the steps that are only relevant to test the quality of the solution approach

5.1.2 Convex Hull Method

In 2010, Aguilera et al (2010) proposed an alternative approach to regressions performed on data known to have a convex shape. This method is simple and fast to model, and can be applied in any dimension and to any convex function. In general, the method consists of a data smoothing step and a *convexification step*. Aguilera et al. (2010) obtained uniform error estimates and showed that the convexification step adds no further errors to the estimation step.

Method

For the remainder of the thesis we assume, without loss of generality (w.l.o.g.), that the data set H is ordered with respect to the first coordinates, i.e. H is such that $x_1 \leq x_2 \leq ... \leq x_h$. The range of x_k is set to the closed interval $Q = [x_1, x_h] \subset \mathbb{R}$, where x_1 and x_h denote the first and last element of the ordered set x. Hence, x_1 and x_h are the minimum and the maximum value of x in the data set respectively.

We assume $f \in \Phi$, where Φ is the set of, finite real valued, non-decreasing concave functions defined on Q. We then define f_h as a naive, point-wise estimator of f defined on Q. This f_h will be derived via a smoothing procedure, i.e. a Local Weighted Scatterplot Smoothing (LOESS) algorithm, on H. Based on f_h a concave estimator $\hat{f}_h$ will be created via an analogue concave adaptation of the *convexification step* as presented in Aguilera et al.(2010). This concave function $\hat{f}_h$ is our final estimator of the real function f. The derivations of both f_h and $\hat{f}_h$ are explained into more detail in the following paragraphs.

To create the initial estimator f_h, a Local Weighted Scatterplot Smoothing (LOESS) algorithm is used, as suggested in Aguilera et al. The LOESS algorithm was first proposed in Cleveland (1979), and is an extension to the local fitting of polynomials as presented in Macauley (1931). The LOESS procedure is often viewed as a vertical sliding window moving over an (x,y)-plot of the data. At every predetermined stopping point the algorithm fits a polynomial through the data points included in the window, according to the standard least-squares measure. This new polynomial is then used to determine an estimated value for the stopping point. Hence, the algorithm brings forth a set of estimated regression values for a set of stopping points determined beforehand.

In order to initiate the algorithm m evaluation points, κ_j for j=1,...,m need to be chosen. We take every x-coordinate from H as an evaluation point ensuring a similar spread in estimations as in the original data set, implying m=h. It then directly follows that the domain of κ equals the domain of Q, where κ denotes the set of evaluation points. We define $(x_{\kappa_j}, \hat{y}_{\kappa_j})$ as the set of points included in the local weighted regression, with the number of points included equal to $\lambda_1 * |H|$ and weighted with predefined weight function Λ. The local regression is a polynomial and so is shaped as

$$f_{\kappa_j} = \sum_{l=0}^{\lambda_2} a_l \cdot x_{\kappa_j}^l$$

with constant $a \in \mathbb{R}$ and predetermined $\lambda \in \mathbb{N}$ as the maximum degree of the polynomial. The estimated y-value for κ_j then is $f_{\kappa_j}(\kappa_j)$. In conclusion, the initial estimator $f_h = \{f_{\kappa_j}\}_{j=1,..,m}$, i.e. the set consisting of every point estimate of the LOESS algorithm.

To implement this algorithm and derive f_h the parameters λ_1, and λ_2 and weight function Λ need to be properly defined and assigned a fitting value. These are stated as:

λ_1: Fraction Data Included

The constant $\lambda_1 \in (0, 1]$ determines the fraction of data points to be considered in each local regression of the LOESS algorithm. Thus, for a local regression at evaluation point κ_j, the $\lambda_1 * |H|$ closest data points w.r.t. κ_j are incorporated in the regression. This implies that for $\lambda_1 \to 0$ only the point closest to κ_j, i.e. in this thesis it would be κ_j itself, is considered. Thus, every data point (x_k, y_k) would be estimated by only taking into account that exact data point (x_k, y_k). Hence, the estimator f_h would equal the data set H.

Similar arguments yield that for $\lambda_1 = 1$ all datapoints are included in every local regression. This would yield a more smooth regression as opposed to $\lambda_1 \to 0$. This holds in general; for larger values of λ_1, a more smooth regression follows. Please note that for $\lambda_1 = 1$ not all κ_j yield the same estimator value. The difference in estimators is a consequence of the predetermined weight function Λ, described into more detail in an upcoming section. This weight function assigns a value to each data point in the regression, representing the importance of the fit of the regression at this point. In general, this weight function is inversely related to the distance of the considered point to κ_j.

As prescribed in Cleveland (1979) the value of λ_1 should in general be between .2 and .8. A value lower than .2 would generate a too inconsistent estimator, and a value higher than .8 would in general produce results which are not able to capture the shape of the data. In this thesis the value is set at $\lambda_1 = .4$. The reasoning behind this is two-fold, and the consecutive fine-tuning is done via experimentation. Firstly, the value of λ_1 should be less than .5, because the details of the local shape are of importance to capture within the model, Cleveland (1979). Since we are trying to fit in a concave shape, this local behavior of the data matters. Secondly, $\lambda_1 > 0.3$ should hold because of the possible impact of a peak in our estimator f_h on the final $\hat{f}_h$ as described in Cleveland (1979). A relatively large peak in the estimator could alter the shape of the final estimator to large extends. This follows from the convex hull taken of f_h to create $\hat{f}_h$. This procedure is described into detail in the following sections, and will give more insight into and more body to this argument.

λ_2: Maximum Degree of Polynomial

For every κ_j a local polynomial is fitted according to the least-squares measure. The parameter λ_2 determines the maximum degree of this local polynomial. In Cleveland (1979) it was issued that $\lambda_2 \geq 2$ was computationally heavy and that $\lambda_2 = 2$ would suffice in the majority of cases. Since computers have evolved drastically since 1979, computational complexity is much less of a burden, and $\lambda_2 = 2$ now provides a balanced choice between computational complexity and the ability to capture a good estimator for the data, as is stated in Jacoby(2000). The dominant improvement of $\lambda_2 = 2$ as opposed to $\lambda_2 = 1$ is to capture curvature of the data more accurately in f_h.

Λ: Weight-Function

In every local regression, the points included are weighted according to the weight-function Λ. This ensures that points closer to κ_j are more important to fit the regression than those farther away. In Cleveland (1979) it is issued that the tricube function, i.e.

$$\Lambda(u_k) = (1 - u_k^3)^3 \text{ for } 0 \leq u \leq 1$$

where u_k is the relative distance from x_k to κ_j, is the best choice for Λ. Please refer to Cleveland(1979) for more details about the requirements for a weight function and the arguments for the choice of the tricube function.

After f_h is determined, a set of points M_h is drawn from f_h. Aguilera et al (2010) state that, in general, the number of points in M_h need not equal h, and the points in M_h might be completely unrelated to the first coordinates of the data points (x,y). The best choice for M_h is dependent on size of the data set, distances between pairs of data points and computational complexity. In this thesis we select every other point and the first and last point in f_h, such that M_h also spans Q. From trial and error it follows that no significant loss of precision from the estimator occurs, paired with a decrease in computational complexity. This particular choice for M_h is further motivated by the automatic emphasis placed upon the parts of the data where more information is present in the data set. Due to nonuniform distributed data across the x-axis, some parts provide more evidence regarding the shape of the data, while stronger assumptions regarding linearity are made on the parts where data is more scarce. Hence, this definition of M_h focuses more on the information-dense parts and leads to a more evidence-based determination of the estimator $\hat{f}_h$.

The following definitions and derivations are parallel to Aguilera et al. (2010), but used to derive a concave function as opposed to a convex function. Let Φ again denote the set of all non-decreasing concave functions, and

$$\Phi_h = \{\phi \in \Phi : \phi(z) \geq f_h(z) \; \forall z \in M_h\}$$

denote the set of all concave functions on M_h which lie above f_h. Then we can define

$$\hat{f}_h = \inf\{\phi : \phi \in \Phi_h\}$$

as our concave estimator on the domain of Q. Since M_h spans Q it is readily seen that $\hat{f}_h$ is well defined on Q.

The procedure of this algorithm for one campaign is presented in Figure 3. In plot b. the LOESS regression on the data points is shown. As stated before, the fractional proportion of data used to estimate each data point is equal to .4. Plot c. shows M_h, which is the union of every other point on the LOESS regression as seen in plot b. and $f_h(x_1)$ and $f_h(x_h)$. Plot d shows the Convex Hull taken over M_h, where in plot e. the desired infimum is shown, which is the red line. Thus, this red line represents $\hat{f}_h$, which also appears in plot f. relative to the original data points. The whole procedure can be seen in one plot in plot g.

5.1.3 Quadratic Formulation

In Chapter 3 it was already stated that from Bhattacharya (2010) and Bhattacharya (2012) we know that, per campaign, advertisement costs should have a monotonically non-decreasing concave relation with generated revenues. We could model this directly as a least squares regression model with the described shape. The model is set up in the following way.

The Model

The input for this model is again the set of history data H. The objective is to create a monotonically non-decreasing concave well-fitted regression line $\tilde{f}_h$ through H. We start out by dividing the range of x into n-1 equally sized intervals, leading to n gridlines. Let $p_i = (i-1) \cdot \frac{x_h - x_1}{n}$ denote the x-value of the i'th gridline, i.e. the i'th gridpoint. Let q_i denote the y-value corresponding to p_i for i=1,...,n , being the decision variable of the model. We assume linear behavior of the regression line $\tilde{f}_h$ between gridpoints. Thus, the set (p, q) denotes the corner points of $\tilde{f}_h$.

Since the regression is defined for the complete set Q, it follows that for every (x_k, y_k) there exists an i such that $p_i \leq x_k \leq p_{i+1}$ holds. Thus, every x_k can be written as a convex combination of p_i and p_{i+1} for a certain i. We denote this as

$$x_k = u_k \cdot p_{i^*} + (1 - u_k) \cdot p_{i^*+1}$$

with $u_k \in [0, 1]$ and $p_{i^*} = \max_j p_j \leq x_k$. It follows that

$$u_k = \frac{p_{i^*+1} - x_k}{p_{i^*+1} - p_{i^*}}$$

and that

$$\tilde{f}_h(x_k) = u_k \cdot q_{i^*} + (1 - u_k) \cdot q_{i^*+1}$$

The objective is to minimize the total distance between $\tilde{f}_h$ and the data points in H. Let d_k denote the distance of point (x_k, y_k) to $\tilde{f}_h$. Then it holds that

$$d_k = |y_k - \tilde{f}_h(x_k)| = |y_k - (u_k \cdot q_{i^*} + (1 - u) \cdot q_{i^*+1})| \quad \forall k \tag{8}$$

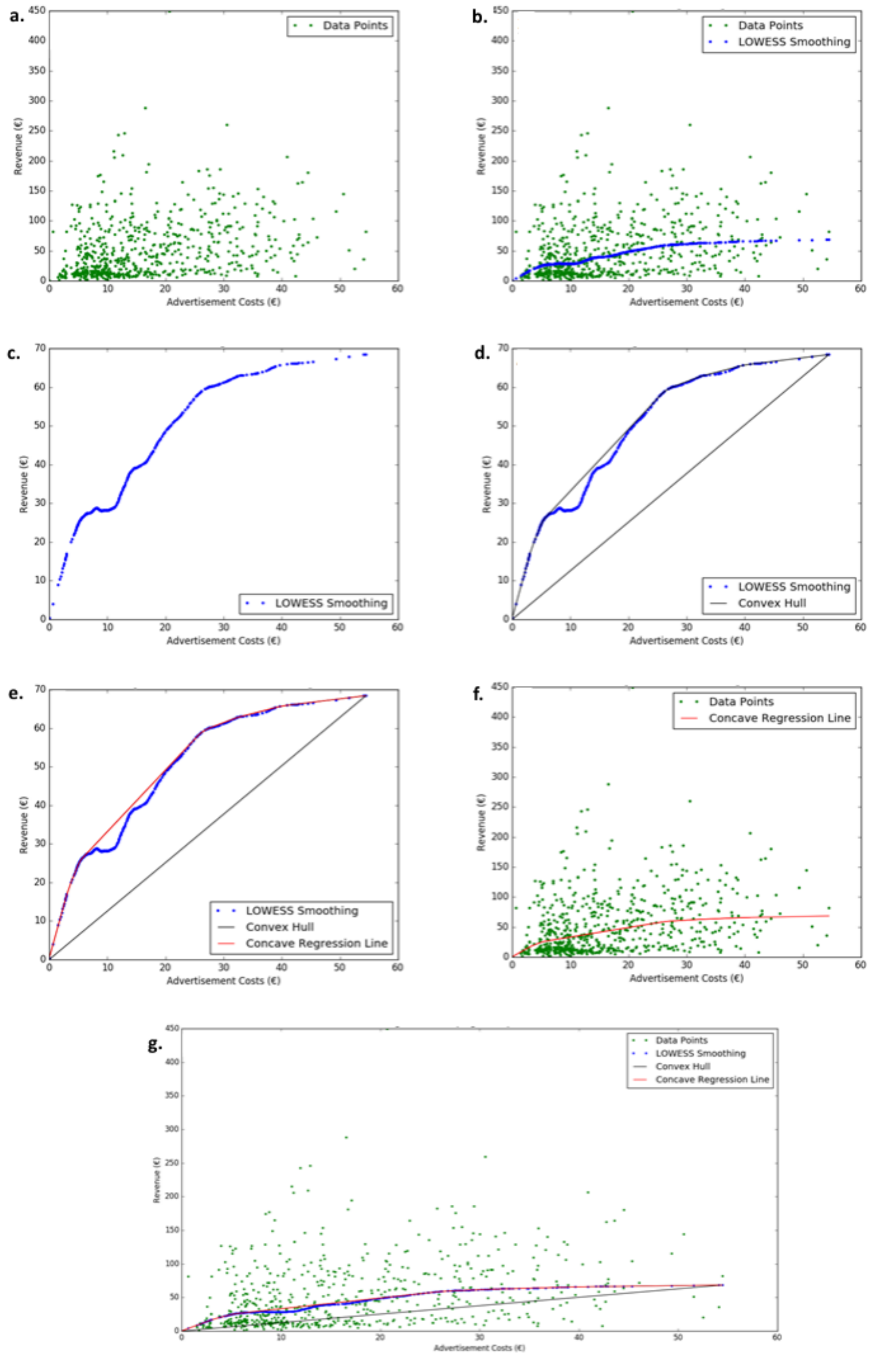

Figure 3: Steps a. through f. of Convex Hull Method, with g. being a complete overview. The separate plots show: a. Data points, b. LOESS smoothing through the data points, c. LOESS regression points, d. LOESS regression points with corresponding convex hull, e. LOESS regression points with concave regression line, f. Concave regression through the data points, g. A complete overview

The constraints of this model concern the general shape of the resulting $\tilde{f}_h$. Let $\pi_i = \frac{q_{i+1}-q_i}{p_{i+1}-p_i}$ denote the slope from gridline i to gridline i+1. To ensure a monotonically non-decreasing $\tilde{f}_h$ we need $\pi_i \geq 0$ to hold for all i. Concavity is then ensured by imposing $\pi_i \leq \pi_j$ to hold for every $j \leq i$, yielding decreasing slope-values in i. In addition to these shape constraints, we also define $q_0 = 0$ to ensure the assumption of a zero profit from zero investment.

The model has the following Second Order Cone formulation:

$$
\begin{aligned}
\text{(SOCP)} \quad \min \quad & ||d|| && && (9)\\
s.t. \quad & q_0 = 0 && && (10)\\
& q_i \leq q_{i+1} && \forall i = 1,..,n && (11)\\
& (q_{i+1} - q_i) \leq (q_{j+1} - q_j) && \forall j \leq i && (12)\\
& d_k = y_k - (u_k \cdot q_{i^*} + (1 - u_k) \cdot q_{i^*+1}) && \forall k \in K && (13)
\end{aligned}
$$

In this Second Order Cone formulation equation (9) and (13) jointly follow directly from (8). Constraint (10) depicts the zero-investment leads to zero-profit assumption and constraints (11) and (12) represent the monotonic non-decreasing concave nature desired of $\hat{f}_n$. The latter two constraints follow from

$$\pi_i = \frac{q_{i+1} - q_i}{p_{i+1} - p_i} \geq 0 \Leftrightarrow q_i \leq q_{i+1}$$

and

$$\pi_i \leq \pi_j \Leftrightarrow \frac{q_{i+1} - q_i}{p_{i+1} - p_i} \leq \frac{q_{j+1} - q_j}{p_{j+1} - p_j} \Leftrightarrow (q_{i+1} - q_i) \leq (q_{j+1} - q_j)$$

since

$$p_{i+1} - p_i = ((i+1) - 1) \cdot \frac{x_h - x_1}{n} - (i - 1) \cdot \frac{x_h - x_1}{n} = \frac{x_h - x_1}{n}$$

is a constant for all i.

The model is presented graphically in Figure 4.

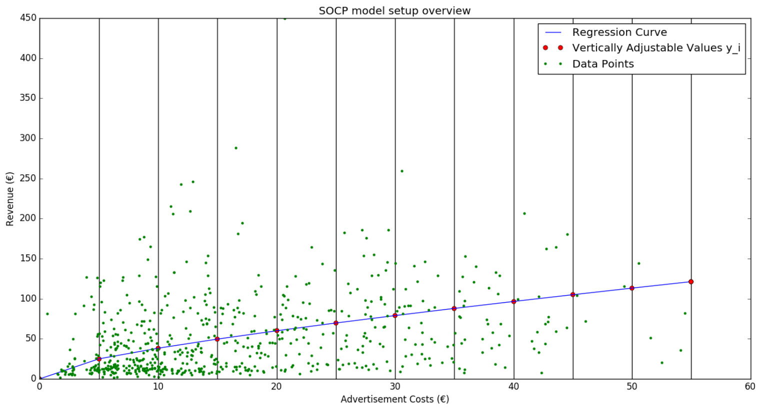

Figure 4: Graphical representation of SOCP model for a concave regression

5.2 Data Uncertainty: Advertisement Costs

The concave regression methods used to establish the expected-sales curve are performed in a deterministic setting. This implicitly assumes the data is correct and insusceptible to uncertainty. This is not an accurate assumption in practice.

The end-user of the Adwords campaigns can assign a budget to be invested in the different campaigns. However, the true expenditure is ultimately determined by Google Inc. as described in Google Support(June, 2016). To fit the needs of his clients as well as possible, a large number of estimations and assumptions are being made by Google Inc.. This leads to a margin of error, for which they account by stating that, on a daily basis, Google Inc. is allowed to spend up to 20% above budget.

The lower bound of the true expenditure by Google Inc. is in theory fixed to 0. Because, when none of the specified keywords are used in the search engine, no bid will be placed and no advertisement will be bought. Since this is a very extreme and exceptional case, a more realistic lower bound to be assumed here is an under-expenditure up to 20% as well. This bound of 20% is a rough estimation from the past experiences at PauwR. In short, if the end-user assigns an amount b_c to a campaign, the actual investment is expected to be in $[.8{\cdot}b_c, 1.2{\cdot}b_c]$.

In accordance with these characteristics of the uncertainty regarding the expenditure parameter, the advertisement costs of each data point k can be reformulated, similarly as in Bertsimas and Thiele (2006), as

$$\bar{x}_k = \bar{\alpha}_k \cdot x_k + \tilde{\alpha}_k \cdot x_k \cdot \alpha_k$$

with $\bar{\alpha}_k = 1$, $\tilde{\alpha}_k = .2$ and $\alpha_k \in [-1, 1]$ an uncertain parameter for all k. This results in $\bar{x}_k = x_k + 0.2 \cdot x_k \cdot \alpha_k$. With this definition of our data points we can setup a robust counterpart for the SOCP formulation of the concave regression. The robust counterpart solves the concave regression problem for the worst-case scenario of the data points.

Please note that, from this point onward, the Convex Hull Method will not be considered as a solution method when considering robust solutions for the data uncertainty. The reason for this being the computational time and storage space of the CHM being much larger than the SOCP formulation. With the SOCP formulation already using the available hardware to its full potential, the CHM results in system failure. In addition, with time restraints regarding this thesis, the extensive rewriting of the CHM to make it work would be too time consuming. Moreover, since, as is seen in Chapter 7, the CHM is already inferior to a great extend compared to the SOCP-formulation when applied to the real-life data sets, the SOCP-formulation is chosen to continue with.

5.2.1 Reformulation Nominal Quadratic Formulation

To write the robust counterpart of the SOCP presented in the Section 5.1 in an understandable way, we first rewrite the nominal SOCP. This is done in the following way.

First note that every x_k can be written as:

$$x_k = \sum_i u_{ik} p_i \tag{14}$$

$$\text{Where } 0 \leq u_{ik} \leq 1 \qquad \forall i, k \tag{15}$$

$$\sum_i u_{ik} = 1 \qquad \forall k \tag{16}$$

$$\sum_i |\mathbb{1}_{u_{i,k}>0} - \mathbb{1}_{u_{i-1,k}>0}| \leq 2 \qquad \forall k \tag{17}$$

Where $u \in \mathbb{R}^{n \times h}$ and $\mathbb{1}_{u_{ik}>0}$ denotes the number of times that a $u_{ik} > 0$. Hence, equation (14) gives a more general description of x_k, allowing it to be met by any combination of the different p_i. However, constraints (15) and (16) ensure that this is a convex combination. Moreover, constraint (17) ensures that for all $k \in K$ at most 2 $u_{ik} > 0$, i.e. always 2, except when $x_k = p_i$ for a k. More specifically, constraint (17) ensures that the positive u_{ik} need to correspond to adjacent p_i. With these constraints the definition of u_k is analogue with the first representation of the SOCP.

We can now define the set U as:

$$U = \{u : (14) - (17)\}$$

Which denotes every possible u, denoting combinations of u_k that satisfy constraints (14)-(17). Note that, in the nominal case, these u_{ik} are unique, due to the deterministic nature of x_k. Thus, U only contains one element; a unique matrix u.

This implies that the nominal SOCP can be equivalently represented by

$$
\begin{array}{lll}
\text{(SOCP)} & \min & ||d|| \\
& s.t. & q_0 = 0 \\
& & q_i \leq q_{i+1} \qquad \forall i = 1,..,n \\
& & (q_{i+1} - q_i) \leq (q_{j+1} - q_j) \qquad \forall j \leq i \\
& & d_k = y_k - (u_k \cdot q) \qquad \forall k \in K, u \in U
\end{array}
$$

5.2.2 Robust Quadratic Formulation

Recall the variable $\bar{x}_k$, presented in Section 5.2.1 as:

$$\bar{x}_k = x_k(1 + 0.2 \cdot \alpha_k)$$

With $\alpha \in \mathcal{A}$, and

$$\mathcal{A} = \{\alpha : -1 \leq \alpha_k \leq 1 \ \ \forall k; \sum_k |\alpha_k| \leq \Omega\}$$

Note that this definition is directly implemented from Bertsimas and Thiele (2006) and that it is a simple mathematical representation of the definition stated in Section 5.2.1.

Since the new data points $\bar{x}_k$ now depend on α, the new definition of U becomes:

$$U = \{u : \exists \alpha \in \mathcal{A}, \bar{x}_k(\alpha_k) = \sum_i u_{ik} \cdot p_i \ \ \forall k; (15) - (17)\}$$

With this new, uncertain variant of the set U, we can easily reformulate the SOCP model to its robust counterpart in the following way:

$$
\begin{array}{lllr}
\text{(SOCPr)} & \min & \sum_k d_k^2 & (18) \\
& s.t. & q_0 = 0 & (19) \\
& & q_i \leq q_{i+1} \qquad \forall i = 1,..,n & (20) \\
& & (q_{i+1} - q_i) \leq (q_{j+1} - q_j) \qquad \forall j \leq i & (21) \\
& & d_k \geq |y_k - (u_k \cdot q)| \qquad \forall k \in K, \forall u \in U & (22)
\end{array}
$$

Where (22) is equivalent to

$$d_k \geq \max\{|y_k - (u_k \cdot q)|\} \quad \forall k \in K, \forall u \in U \tag{23}$$

Note that in equation (22), the equality is replaced by an inequality with absolute values and consequentially the norm in (18) is replaced with a squared sum. This follows directly from the fact that (22) needs to be equivalent to (23). Otherwise stated, we want (22) to ensure that d_k equals the expression in the worst-case scenario w.r.t. u. Hence, the equality needs to replaced with a greater equal sign such that, in optimality, the d_k is equal to the maximum value of the expression. As a consequence, the absolute values need to be placed, such that we only consider the distance between the two expressions. In the original SOCP this was automatically taken care of by the norm in the objective, however this trick cannot be carried over to the robust problem. This then causes (18) to be reduced to a sum statement.

From (23) it is readily seen that we are dealing with a maximization subproblem within a minimization problem. In Bertsimas and Thiele (2006) they tackled this problem by first extracting the maximization subproblem. They then calculated the Lagrangian Dual for (23), and concluded that the Slater Condition for strong duality was met and therefore that at optimality the solutions of the dual and primal would be equal. This enabled them to replace the maximization primal by the minimization dual in the robust problem, without affecting the solution of the robust problem. Hence, they ended up with a minimization subproblem inside a minimization problem, i.e. the minimization statement in the subproblem ceased to exist and the general problem became solvable. For a more in-depth explanation about this methodology please refer to Bertsimas and Thiele (2006).

Even though a lot of parallels can be drawn from the problem tackled in Bertsimas and Thiele (2006) and the SOCP at hand, there is one particularly tricky difference. This difference is the combinatorial complexity of the set U. This prohibits the use of the technique presented in Bertsimas and Thiele(2006). This prohibition follows directly from the inability to calculate a closed form Lagrangian Dual for the (23). Consequently, there is no known exact method to solve the problem at hand, and thus the focus will be shifted towards the usage of heuristics. The heuristic used in this thesis is known as the Adversarial Approach, and will be explained into more detail in the upcoming sections.

Please note that, despite the inability to implement the method of Bertsimas and Thiele (2006) in general, there are exceptions. More specifically, the method of Bertsimas and Thiele (2006) can be implemented if one or both of the following conditions are met:

1. **When no $\bar{x}_k$ intersects with a gridline.** This ensures that the u_{ik} values which are positive are the same as with the nominal case. This implies that the only extra variance obtained is the varying values of the two positive values of u_{ik}. Hence, the problem becomes much easier to solve.

2. **When no $\bar{x}_k$ intersects with the regression line.** When it is known beforehand which points will be below the regression line, and which will be above it, the problem can be rewritten into a much easier to solve format. Because this condition states which direction the points will be moved in the worst-case. This implies that all points above the line are tending to the left, where the lines below the line have the tendency to move right, w.r.t. the nominal solution. So the only important added variable remains to determine which $\bar{x}_k \neq x_k$ in the worst-case. Thus, again the problem becomes much easier to solve.

5.2.3 Adversarial Approach

The adversarial approach is, as stated in Chapter 3, a heuristic used to solve robust counterparts. This approach was introduced by Bienstock and Ozgul (2008). The general idea is to only consider the elements of the uncertainty set that are important to the problem, as opposed to considering the uncertainty set as a whole. Hence, only elements $\bar{u} \in U$ which are candidates to be the worst-case scenario. As a matter of notation we denote $\bar{U} \subset U$ to be the relevant subset of U when determining the robust solution.

In the adversarial approach $\bar{U}$ is build-up iteratively, starting with the nominal case. Every iteration of the algorithm the robust model is solved for uncertainty set $\bar{U}$. Based upon the current solution, an adversary determines the worst-case scenario to occur with respect to the data points. Hence, the adversary determines an $\bar{u} \in U$ that is the worst-case for the currently proposed solution of the algorithm. This $\bar{u}$ is then added to $\bar{U}$, and the robust model is solved again, for an extended set $\bar{U}$. This repeats until a certain stopping criterion is met. The algorithm is explained into more detail in the upcoming sections.

Adversarial Algorithm

The algorithm can be divided into 5 sequential steps with the following contents:

Step 0

Define a stopping criterion s
Set the nominal value x^0 and corresponding u^0
Set iteration parameter $\iota = 0$

Step 1

Define the complete uncertainty set U defined as in (5.2.2). Hence, U is the set containing all possible vectors u given the budget constraint.
The algorithm starts with the uncertainty set $\bar{U} = u^0$, which is the set of parameters corresponding to x^0.

Step 2

Solve the nominal problem:

$$
\begin{aligned}
\text{(SOCPr)} \quad \min \quad & \sum_k d_k \\
s.t. \quad & q_0 = 0 \\
& q_i \le q_{i+1} && \forall i = 1,..,n \\
& (q_{i+1} - q_i) \le (q_{j+1} - q_j) && \forall j \le i \\
& d_k \ge |y_k - (\bar{u}_k * q)| && \forall k \in K, \forall \bar{u} \in \bar{U}
\end{aligned}
$$

And obtain solution $\hat{q}$, resulting in $\beta_\iota = ||d||$.

Step 3

For obtained $\hat{q}$, solve the adversarial problem:

$$\begin{aligned} \max \quad & \sum_k |y_k - (u_k * q)| \\ s.t \quad & u \in U \end{aligned}$$

And obtain solution $\tilde{u}$.

Step 4
IF β_ι-$\beta_{\iota-1} > s$
Then $\iota \leftarrow \iota + 1$
$\bar{U} = \bar{U} \cup \tilde{u}$
Go back to step 2
ELSE: Stop

The adversarial problem to be solved in step 3 is depicted in figure 5. We can divide the data points to be evaluated by the algorithm, roughly into three different categories. The first category consists of the data points of whom the uncertainty set lies completely above the regression curve determined in Step 2. For these data points the interesting part of the uncertainty interval is the left-hand side, because of the monotonically non-decreasing property of the curve. The second category is the antagonist, i.e. the uncertainty sets lying wholly underneath the regression line. For these points the interesting part is the right-hand side. The last category is the the group of data points for which the uncertainty interval intersects with the regression curve. For these points both the right-hand and the left-hand side could prove to be optimal.

When no budget of uncertainty would be imposed, or when this budget is sufficiently large, all data points would move to one of the ends of the uncertainty interval, i.e. $|\alpha_k| = 1 \; \forall k$. This is again because of the monotonic non-decreasing shape of the regression curve. When an interval lies completely above the line, the objective value for the adversarial could never become worse when moving to the left. A similar argument holds for intervals lying beneath the curve with moving to the right. The last category also ends up in one of the extremes of the interval, because in this reasoning you could split-up the interval in the intersection point. Hence, you have two intervals, one from category one and one from category two. These maximums are in the extremes, and hence the maximum of these two extremes is also the global maximum for the interval in total.

Implementation Adversarial Algorithm
In Step 0 the stopping criterion is defined as $s = 0.01$, and the nominal value u_0 is equal to the value of u derived in the nominal SOCP.

We start with the nominal set of data points and corresponding u in step 1. Step 2 is by definition similar to the nominal case of solving. With the single exception that every iteration of the algorithm a new set of constraints is added, due to the extension of $\bar{U}$.

In step 3, we have a budget of uncertainty, given in step 1. This budget is to be spread in an efficient way, such that we get the worst possible data set

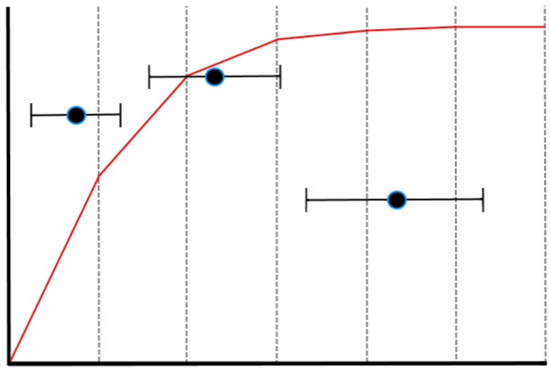

Figure 5: Schematic overview of uncertainty in the x-values

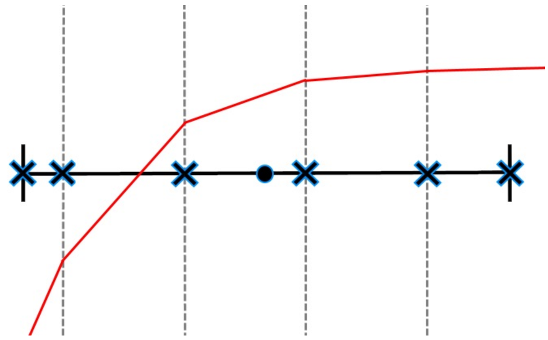

Figure 6: Example of Points of Interest for uncertain (y_k, x_k)

with respect to the current regression curve. Determining this data set is done via a cost-savings algorithm approach.

Firstly, for every $\bar{x}_k$ the points of interest (PoI) are derived. These points are depicted in Figure 6. The points include the extremes of the intervals of uncertainty, as well as the nominal value x_k and the intersections of the intervals of uncertainty with the gridlines. The extremes are interesting because these are the points that would be chosen when there was sufficient budget, because these lead to the greatest value for the adversarial. The nominal value is interesting because this is the starting point for each $\bar{x}_k$ and the savings of the adversarial are calculated with respect to the base level. Finally, the intersection with the gridlines are interesting because of the piece-wise linearity of the regression curve. Ultimately, the adversarial wants to consider every possible value of α to know which will give him the biggest savings. However, because of the linear behavior of the regression curves in between gridlines, the ratio between costs and savings is constant over this part of the interval.

Then, for every PoI for all $\bar{x}_k$, the following values are tabulated

α	sign	$\lvert\alpha\rvert$	increment	ratio($\frac{incr}{\lvert\alpha\rvert}$)	k
$\in [-1, 1]$	$\in \{-1, 1\}$	$\in [0, 1]$	$\in \mathbb{R}$	$\in \mathbb{R}$	$\in [0, h]$

A certain budget is set, representing the amount of uncertainty the user wants to hedge against. This is set as being a certain percentage of the maximum (equal to the number of data points). In the algorithm, this budget is spend on the different x_k using an iterative approach. Let Ω denote the total budget of uncertainty and let ω denote the remaining budget of uncertainty.

At each iteration the algorithm chooses the entry in the previously specified PoI-matrix, with the maximum ratio of increment compared to the cost $|\alpha_k$. Say it picks entry ρ with values $\tilde{\alpha}$, $\widetilde{sign}$, $\widetilde{incr}$, $\widetilde{ratio}$ and $\tilde{k}$. Then, the following variables are altered as:

- $\bar{x}_{\tilde{k}} \leftarrow \bar{x}_{\tilde{k}} + \widetilde{incr}$
- $\omega \leftarrow \omega - |\tilde{\alpha}|$
- **For all PoI where $k = \tilde{k}$:**
  if $|\alpha| \leq |\tilde{\alpha}|$:
  entry is removed, because it is less beneficial than the currently selected improvement and will therefore never be selected later on in the algorithm

  else:
  $|\alpha| \leftarrow |\alpha| - |\tilde{\alpha}|$
  $\alpha \leftarrow |\alpha| * sign$
  $incr \leftarrow incr - \widetilde{incr}$
  ratio recalculated with new values of incr and α

Please note that the determination of whether an adjustment needs to take place depends on the absolute value of α. Since (when the possible value line of x_k crosses the regression line,) the algorithm may move to the right first and later to the left to make maximum improvements from the adversarial point of view.

When ω becomes significantly low, i.e. smaller than 1, it may be the case that some points of interest may not be reachable anymore. When this happens the algorithm corrects as is depicted in the Figure 7. This means that points which are not reachable anymore are removed, and replaced by points where α equals the remainder of ω. Corresponding values for the increments are calculated, along with the ratios. These new rows are added to the table, and the algorithm continues until ω reaches zero.

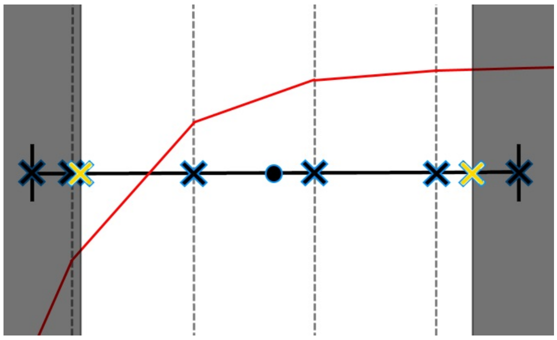

Figure 7: Example of points of interest for uncertain (y_k, x_k) for small level of ω

5.3 Data Uncertainty: Revenue

In addition to the data uncertainty caused by spending policies of Google Inc., there is also uncertainty created by measurement inaccuracies. These inaccuracies have two main origins; time-lags and inter-channel relations. These two phenomena are accounted for with the practical implementations on the PauwR data sets, but only based upon estimations and general trends.

Time-lags follows from customers' thinking time regarding a purchase of a product. This includes the time between the first click on an advertisement by a customer and the moment when the actual purchase occurs. The available information in practice about time-lags are only averages per client of PauwR. This implies that the implemented time-lags are the same for each campaign of a client. They are implemented as fixed percentages of revenue per data point. This is in contrast with reality, where every purchase has an individual lead time. Hence the generation of uncertainty following this effect.

The second phenomenon, the inter-channel relations, encompass carry-over effects from one channel to another. Hence, this captures the acquisitions done via a channel that differs from the original advertisement that lured the customer to the website. This implies that this more often occurs when there is also a significant time-lag. The data available is per investment, so every data point can be individually corrected. Hence, every revenue value can be increased or decreased depending on the data. The uncertainty follows from possible measurement errors or measurement impossibilities resulting in deviated outcomes.

5.3.1 Box uncertainty

Due to the two phenomena described, the uncertainty approach can be extended to two-dimensional uncertainty. Otherwise stated, the interval uncertainty over the x-axis has a similar equivalent on the y-axis caused by the two factors described. We assume that these uncertainties are independent, since the x-uncertainty follows from expenditure decisions made by Google Inc., where the y-uncertainty follows from data flaws and faulty data.

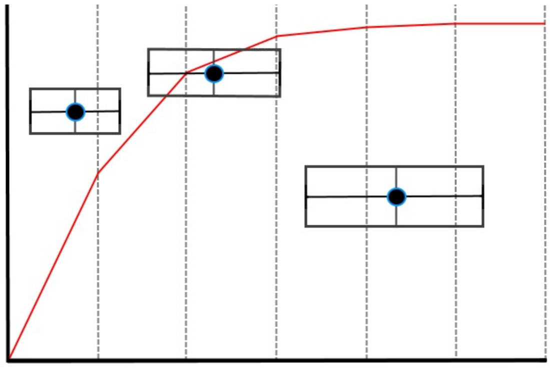

Figure 8: Schematic overview uncertain data points

We do assume the two uncertainties to share a single budget of uncertainty. Hence, we now have the budget constraints

$$\sum_k \gamma \cdot \alpha_k + (1-\gamma) \cdot \beta_k \leq \Omega \tag{24}$$

$$\gamma \in [0,1] \tag{25}$$

$$\alpha_k \in [0,1] \qquad \forall k \tag{26}$$

$$\beta_k \in [0,1] \qquad \forall k \tag{27}$$

$$\tag{28}$$

Where α and Ω are the same as with the data uncertainty case, and β is defined for the y-uncertainty, similarly as α for the x-uncertainty. However, a user-specified value of γ is introduced, which determines the relative costs of α and β.

The shape of the uncertainty is henceforth assumed to be a box. This seems to be the most intuitive shape, due to the presumed independence and thus the realistic possibility for a point to end up in the extreme case for both parameters. The current situation is illustrated in Figure 8.

5.3.2 Adversarial Approach

The main idea behind the Adversarial Approach for this extended case remains the same as before. The major change now lies in the evaluation of 3 times as many points as before. The evaluation points of the previous, more simple case remain but are accompanied with 2 sets of points on the extremes of the y-axis as well. This is represented in Figure 9.

Even though the evaluation from an algorithmic point of view is simply an extended version from before, the intuition behind the Y-Uncertainty is much simpler. For the boxes entirely above or below the curve, the direction of the y-uncertainty from the adversary would be upwards or downwards respectively. For the boxes that intersect with the regression curve, the side would be chosen

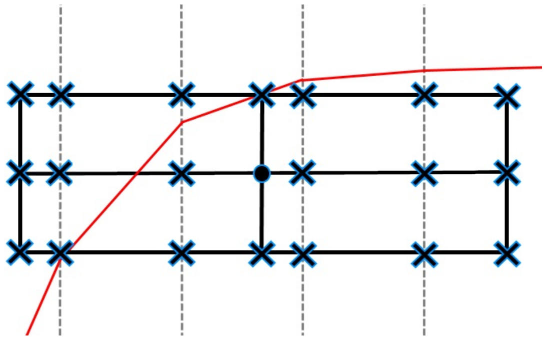

Figure 9: Example points of interest for uncertain (y_k, x_k)

which in which the curve does not intersect with the box. In Figure 9 this would be the upper half if the point selected for the x-uncertainty would be to the left, and the upper-half for the right.

The selection for which box to move to the y-direction is intuitively much easier as well. This is entirely determined by the size of the boxes in terms of y. Due to the definition of the y-uncertainty, the boxes are given differing measures. And since, for given x-uncertainty, the added savings for the adversarial are linear in β, the most y-uncertainty is given to the larger boxes.

The situation described would be a trivial one to solve. However, the extension is made harder to solve for the adversary because he has to evaluate the x and y-uncertainty simultaneously due to their common budget of uncertainty. This leads to the extension by the algorithm as stated before and shown in Figure 9. Particularly, the algorithm remains the same, but with more entries in the presented table, as well as added columns, leading to the following table:

$\alpha \in [-1,1]$	sgn $\alpha \in \{-1,1\}$	$\lvert\alpha\rvert \in [0,1]$	$\beta \in [-1,1]$	sgn $\beta \in \{-1,1\}$	$\lvert\beta\rvert \in [0,1]$	cost $\in [0,1]$	incr $\in \mathbb{R}$	$(\frac{incr}{costs}) \in \mathbb{R}$	$k \in [0,h]$

Where costs is defined according to the previous definition; $costs = \gamma \cdot |\alpha| + (1-\gamma) * |\beta|$.

When $\omega < 1$, the solution approach alters in an analogue variant to before. So, the now excluded points are removed from the table, where the boundary values, i.e. where $\gamma \cdot |\alpha| + (1-\gamma) \cdot |\beta| = \omega$, are added. This is illustrated in Figure 10.

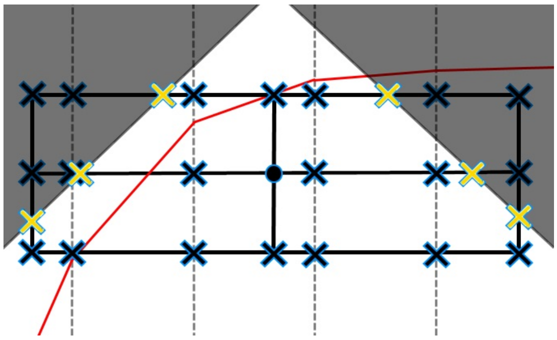

Figure 10: Example points of interest for uncertain (y_k, x_k) for small value of ω

6 Implementation

The solution approach described in the previous chapter is tested on multiple data sets. First, in Chapter 6.1, it is implemented on a real data set, imported from the Google Analytics data base of an anonymous client of PauwR. To further improve the results, Chapter 6.2 discusses how possible outliers are detected and removed in this data set. This is done via a Simulated Annealing algorithm. Lastly, the solution approach is applied to several artificially generated data sets to test the performance of the algorithm and perform sensitivity analyses in Chapter 6.3.

6.1 PauwR Data Set

6.1.1 Data Retrieval

Via the Google Analytics API the data for PauwR's anonymous client X for the period 14-01-2014 until 22-10-2015 are read-in. The data consists of the revenues and costs per campaign for this client. In addition, the revenues from other sources are read-in, as well as the data of 30 days preceding and succeeding the specified time interval. This is done to account for the carry-over effects and time-lags described in Chapter 5 and further illustrated in the next section.

6.1.2 Data Modifications

The data read in from the Google Analytics API is raw data. The data is one-to-one matched costs and revenues. As stated in Chapter 5, we need to account for certain time-lags following customer's consideration times as well as carry-over effects from other marketing channels. The marketing channels considered to have carry-over effects are the direct purchases, i.e. by typing in the URL of the clients website, and organic search links, i.e. the links that are found by the used search engine when customers search for the product they desire. The last category of carry-over effects to be considered are the mutual carry-over effects of the AdWords campaigns. In addition to these effects, a grouping of AdWords campaigns was done for client X. This was done for implementation purposes.

These performed operations in practice are explained into more detail in the following subsections.

Time-lags

Google Analytics solely provides cumulative data on the average consideration time of a customer regarding a purchase from your webshop. Hence, only a general estimation can be made of the time-lags. In this case about 70% of the customers buys the same day as they first clicked on a sponsored link. The remaining 30% purchases within the next 30 days with a negligible loss of data. These daily fractions are applied directly and uniformly to the retrieved data.

Carry-Over

There are customer tracks available in Google Analytics, showing which path customers have followed to purchase the desired product. And how many customers followed this specific path. With this information, for every campaign is filtered out how many started with that campaign and how many purchases ended with what campaign or other marketing channel. This results in ratios of carry-over per campaign, because there is no available data on which exact customers followed the specific paths. The obtained ratios are then applied to the data to backwardly correct for these customers.

Grouping

Since PauwR has used multiple campaigns in the same categories and with similar goals for client X, a grouping is made to create more extensive and complete data sets. Hence, instead of directly using the campaigns as input for the model, a certain grouping has taken place. This grouping would not have been necessary if the campaigns were set up more structured.

The grouping of campaigns can occur due to two different criteria. The first reason to group campaigns is if there were two similar consecutive campaigns. Hence, if there were 2 campaigns in the data set with (almost) identical keywords, and the second campaign began after the first ended. This was in most cases due to the client temporarily stopping with AdWords advertisement in this particular category. The second reason to group was if keywords with similar keywords were run simultaneously. More specifically, this consists of directly competing campaigns and complementary campaigns. This was often due to specific sales done by the client, such that he could get a better overview of the results of these specific sales afterwards.

In conclusion, if multiple campaigns were grouped, these could have, and maybe should have, been in one campaign initially. They were only separated to satisfy secondary needs by the client while not having a need to have differently structured data from PauwR's point of view.

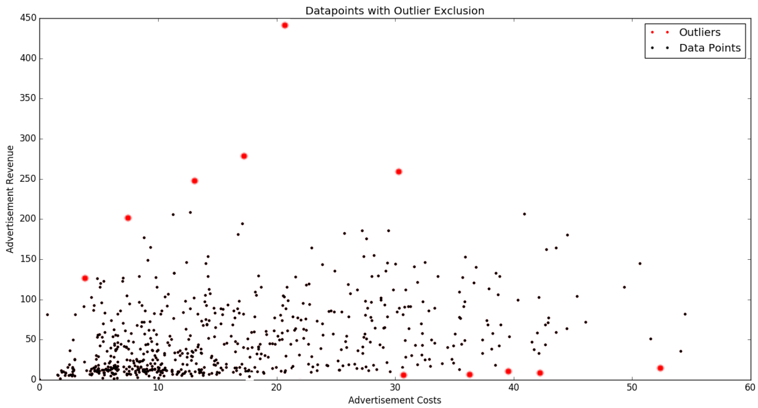

Figure 11: Illustration of outliers (red) to be excluded from dataset (black)

6.1.3 Data Uncertainty

The uncertainty regarding the advertisement costs was stated to be equal to 20% of the investment. This level of uncertainty is the level that Google Inc. states to permit itself. As stated by Google Support (June,2016), these bounds are, in practice, nearly never attained. A more practical uncertainty level, which also leads to less extreme, and hence more interesting, cases is to state that $x_k \in [.9 \cdot x_k, 1.1 \cdot x_k]$. Thus, to assume an uncertainty level of 10% instead of the aforementioned 20%.

The uncertainty regarding the revenues remains unaltered in the practical case. This implies that the fixed proportion of the y-uncertainty is equal to 11%, and the fixed uncertainty added per individual data point lies in the interval $[0, .05 \cdot y_k]$.

6.2 Outlier Exclusion

When dealing with data sets from the real world, encounters with outliers are almost inevitable. The formal definition of an outlier given by Moore and McCabe(1999) is: "An outlier is an observation that lies outside the overall pattern of a distribution". Hawkins described an outlier as an observation that "deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" (Hawkins, 1980). In practice the more extreme outliers are frequently immediately clear from the data. There is however, a large gray area of ambiguity about whether some data points should be in- or excluded from the set, J.W. Osborne and A. Overbay (2004).

Due to the control one has over the advertisement investments (x), the only potential outliers that are of interest in this thesis, are outliers in terms of the dependent variable; revenue (y). See Figure 11 as means of an illustration.

6.2.1 Motivation

According to J.W. Osborne and A. Overbay (2004) outliers can possess a significant power to distort results produced from given data sets. As stated earlier, these outliers seem to follow from a different mechanism than the one investigated. It is important to know the system to conclude about outliers, as stated in J.W. Osborne and A. Overbay (2004).

The practical cases with PauwR is the system of interest in this thesis. In general, these practical cases show many small purchases the majority of the time. However, once every while a big order is placed which orders over 10 times the normal value of an order. This phenomenon is also seen with the most obvious outlier in Fig 11 at a revenue level of 450. For now it seems this investor comes by at random, so we do not account for this in the sales function estimation. This is just seen as a lucky shot.

Since the datasets are relatively small, and the (theoretical) shapes of the data are important, the outliers could influence the results to an excessive extend. This would lead to unwanted results, resulting in a bad model. Hence, outlier exclusion is of importance, and will be used to obtain better results for the stated mathematical model in Chapter 4.

6.2.2 Simulated Annealing: Boundary Creation

Since we limit ourselves to potential outliers in the y-direction, due to the level of controlability over the x-axis, an upper and lower bound are composed through a $100 \cdot (1 - \alpha)\%$ confidence region, formed by a Simulated Annealing algorithm.

Simulated Annealing Setup

Simulated Annealing (SA) is a heuristic, based upon the paper published by Metropolis et al (1953). The algorithm simulates the cooling of material in a heat bath, also known as annealing. While the material is still hot, one can make large adjustments to its shape without being able to do detailed operations. While the material is cooling down, larger parts of the material become solid, leaving room for more precise, minor adjustments. This implies that rough shapes are being formed at the start of the process, slowly converging towards smaller adjustments as the material cools down.

This is translated fairly directly to a mathematical optimization problem of finding a global optimum. At the start of the algorithm, the temperature parameter is high, resulting in a global, more randomized search. In this state, the algorithm allows many uphill moves in terms of the search direction. As the value of the temperature variable is decreasing, the search becomes more locally orientated. When temperature tends to zero, the algorithm only allows for descending movements as a search direction. This ultimately leads to the convergence to a local optimum, which should be the global optimum if the algorithm is setup correctly and the parameter values are chosen well.

Objective and Solution Definition

The SA algorithm in this thesis will determine a set $H^{out} \subset H$, consisting of the data points $(x, y)^{out}$ considered to be outliers. This will be done via a Local Search setup, as is standard for SA. First an initial solution is determined, with corresponding costs and sets H^{out} and $H^{in} = H \setminus H^{out}$. With every iteration a new solution is determined with corresponding costs and sets of points included and excluded. The costs are then determined and whether the new solution is accepted depends on the level of costs and the Metropolis criterion as stated in Metropolis et al. (1953), which will be explained in more detail in the following sections.

Search Space: Neighborhood Definition

SA is classified as a local search algorithm. Local search algorithms move from solution to solution until the optimal solution is found or a specified time bound is reached. In general, local search algorithms are meta heuristics used to solve computationally hard problems, Choi and Lee (1998).

For a local search algorithm to move from one solution to the next, a neighborhood of the solution has to be determined. A neighborhood defines a subspace of the solution space, in which the algorithm searches for the next solution. Let υ denote the current solution of the SA algorithm. Then, according to Dell'Amico et al.(2009), the neighborhood $\mathcal{N}(\upsilon)$ is a function "which associates with any solution υ a portion $\mathcal{N}(\upsilon)$ of the solution space containing all solutions that can be obtained from υ with a 'simple' transformation".

There are many variations of neighborhood functions that can be found in literature. For example, one might base a neighborhood function $\mathcal{N}$ upon a (Euclidean) distance stating that $\mathcal{N}_\delta(\upsilon) = \upsilon' : |\upsilon' - \upsilon| \leq \delta$, where δ is a given constant, as is done in Qian et al. (2004). Another possible categorization of possible neighborhood functions $\mathcal{N}$, more applicable to this case, is the presented in Dell'Amico et al.(2009). This categorization is the following: $\mathcal{N}_1$ (contour filling), $\mathcal{N}_2$ (pairwise-exchange) and $\mathcal{N}_\mathcal{L}$ ($\mathcal{L}$-exchange). In this thesis a combination of contour filling and pairwise-exchange is used, and accordingly, the neighborhood of a solution υ is denoted by $\mathcal{N}_{12}(\upsilon)$.

Say, again, the current solution is υ, with corresponding $H^{out}(\upsilon)$ and $H^{in}(\upsilon)$. Let η^{in} and η^{out} be arbitrary elements in $H^{in}(\upsilon)$ and $H^{out}(\upsilon)$ respectively. Then $\upsilon' \in \mathcal{N}_{12}(\upsilon)$ if and only if $H^{out}(\upsilon')$ and $H^{in}(\upsilon')$ relate in one of the following three ways:

- $H^{out}(\upsilon') = H^{out}(\upsilon) \cup \eta^{in}$ and hence $H^{in}(\upsilon') = H^{in}(\upsilon) \setminus \eta^{in}$
- $H^{in}(\upsilon') = H^{in}(\upsilon) \cup \eta^{out}$ and hence $H^{out}(\upsilon') = H^{out}(\upsilon) \setminus \eta^{out}$
- $H^{in}(\upsilon') = H^{in}(\upsilon) \cup \eta^{out} \setminus \eta^{in}$ and hence $H^{out}(\upsilon') = H^{out}(\upsilon) \cup \eta^{in} \setminus \eta^{out}$

Otherwise stated, a potential candidate solution $\upsilon' \in \upsilon_{12}$ if $H^{in}(\upsilon')$ equals $H^{in}(\upsilon)$ with a possible addition and/or subtraction of single points of the set $H^{in}(\upsilon)$, implying $|H^{in}(\upsilon)| - 1 \leq |H^{in}(\upsilon')| \leq |H^{in}(\upsilon)| + 1$.

Please note that this definition applies to all subsets of the sets of data points, with exception of duplicate points, i.e. points with the same x and y coordinate. These are treated as if they are one point for the sake of the neighborhood definition. They are treated as multiple points for all other purposes, e.g. when considering the confidence levels of the confidence regions and possible penalty costs.

The SA Algorithm

The Simulated Annealing algorithm used to solve the problem at hand is represented in Algorithm 1, and will be extensively discussed in the sequential sections.

Algorithm 1 Simulated Annealing

Initialize (read-in) datapoints H
Set parameters: $T(> t)$,$t(> 0)$,$\delta(\in (0, 1))$,$\rho(> 0)$,$n(> 0)$,$\theta(> 0)$,$\alpha(\in (0, 1))$
Initialize gridlines p_i with i=1,...,n, leading to n-1 intervals
Set $\mathcal{P}[i] = 0$ and $\mathcal{A}[i] = \lfloor (1 - \alpha) \cdot H[i] \rfloor \ for \ i = 1, ..., n$
Initial solution: $\upsilon = [\upsilon_{ub}, \upsilon_{lb}]$
Initial costs: $\mathcal{C}(\upsilon) = \sum(\upsilon_{ub} - \upsilon_{lb})$

while $T > t$ **do**
  $\tau \leftarrow 0$
  $\bar{\tau} \leftarrow 20 \cdot T + 100$
  while $\tau \leq \bar{\tau}$ **do**
    choose random $\upsilon' \in \mathcal{N}_{12}(\upsilon)$
    $\mathcal{P}(\upsilon') \leftarrow \{(H^{out}(\upsilon') - \mathcal{A}[i])\}^+$
    $\mathcal{C}(\upsilon') = \sum(\upsilon_{ub} - \upsilon'_{lb}) + \rho \cdot \sum_i \mathcal{P}[i]$
    $\Delta\mathcal{C} = \mathcal{C}(\upsilon') - \mathcal{C}(\upsilon)$
    draw random $r \sim U(0, 1)$
    if $(\Delta\mathcal{C} < 0 \ or \ e^{-\theta \cdot \Delta\mathcal{C}/T} > r)$ **then**
      $\upsilon \leftarrow \upsilon'$
      $\mathcal{P}(\upsilon) \leftarrow \mathcal{P}(\upsilon')$
    end if
    $\tau \leftarrow \tau + 1$
  end while
  $T \leftarrow \delta * T$
end while

Step 0: Initialization

Choose parameter values for

$T(> t)$: Starting temperature

$t(> 0)$: Temperature threshold

$\delta(\in (0, 1))$: Temperature reduction fraction

$\rho(> 0)$: Penalty costs parameter

$n(> 0)$: Number of gridlines, leading to n-1 intervals

$\theta(> 0)$: Constant for costs scaling

$\alpha(\in (0, 1))$: Level of confidence

The intervals are setup in a similar fashion as with the SOCP model, i.e. with n gridlines uniformly distributed over the interval $[0, x_h]$. The data points are automatically assigned to the interval corresponding to the two enclosing gridlines. We denote the data points in interval i by $H[i]$, with reciprocal gridlines p_{i-1} and p_i. In general these intervals do not contain the same number of data points.

To determine the sets H^{in} and H^{out}, a solution of the SA algorithm states a lower bound and upper bound on H^{in}, i.e. $\upsilon = [\upsilon_{lb}, \upsilon_{ub}]$, with $\upsilon \in \mathbb{R}^{n\times 2}$ and so $\upsilon_{lb}, \upsilon_{ub} \in \mathbb{R}^n$. The entries of υ represent y-values corresponding to p_i for i=1,...,n, analogue to the definition of q with the SOCP model. Moreover, the assumed linearity between consecutive entries applies again. Thus, the sets (p, υ_{lb}) and (p, υ_{ub}) form the corner points of the piece-wise linear lowerbound and upperbound functions, $g_{lb}(\upsilon)$ and $g_{ub}(\upsilon)$, respectively. Then H^{in} is defined as

$$H^{in} = \{(x_k, y_k) \in H : g_{lb}(\upsilon) \leq y_k \leq g_{ub}(\upsilon)\}$$

Note: since $H^{out} = H \setminus H^{in}$, only H^{in} will be extensively treated, with H^{out} implicitly determined.

The initial solution υ_0 is determined such that $H^{in} = H$. This is done by starting at the back-end of the x-axis, p_n, and determining $\upsilon_{lb}[n] = \min\{y_k : (x_k, y_k) \in H, x_k = p_n\}$ and $\upsilon_{ub}[n] = \max\{y_k : (x_k, y_k) \in H, x_k = p_n\}$.Then, g_{ub} is determined using the following iterative procedure for j=0,...,n-1:

1. The current point to evaluate is $(p_{n-j}, \upsilon_{ub}(n-j))$
2. Find $\hat{\sigma}$ such that $\hat{\sigma} = \{\min \sigma | \upsilon_{ub}(n-j) - \sigma \cdot (p_{n-j} - x_\eta) \geq y_\eta \ \forall \eta \in H[n-j-1]\}$. Note that this minimization is due to the fact that the algorithm evaluates from n to 1 backwards.
3. Set new $\upsilon_{ub}(n-j-1) = \upsilon_{ub}(n-j) - \hat{\sigma} \cdot (p_{n-j} - p_{n-j-1})$

The process to determine the initial lower bound is analogue, with an alteration of the definition of $\hat{\sigma}$ to $\hat{\sigma} = \{\max \sigma | \upsilon_{ub}(n-j) - \sigma \cdot (p_{n-j} - x_\eta) \leq y_\eta \ \forall \eta \in H[n-j-1]\}$

After the initial solutions for υ_{ub} and υ_{lb} are calculated, the initial costs are defined by $\mathcal{C}(\upsilon) = \sum(\upsilon_{ub} - \upsilon_{lb})$. The Simulated Annealing Algorithm will start from this solution, and tries to improve the solution over iterations, leading to the best possible approximation of the optimal solution with respect to $\mathcal{C}(\upsilon)$.

Step 1: Search Space (Neighborhoods)
Every iteration of the algorithm, a new solution υ' is proposed, with $\upsilon' \in \mathcal{N}_{12}(\upsilon)$. This new solution is obtained by randomly choosing an interval, a gridline p_i and whether to add an element, subtract or do both. It is also randomly selected whether to do which of the randomly selected operations on the upper or lower bound. In figure 12 the possible options are illustrated for a random interval at a random moment in time.

We denote the set of tight points of interval i by $\mathcal{T}_i$, defined by

$$\mathcal{T}_i = \{\eta \in H_i : \sigma((p_{i-1}, \upsilon_{i-1}), (p_i, \upsilon_i)) = \sigma(\eta, (p_i, \upsilon_i))\}$$

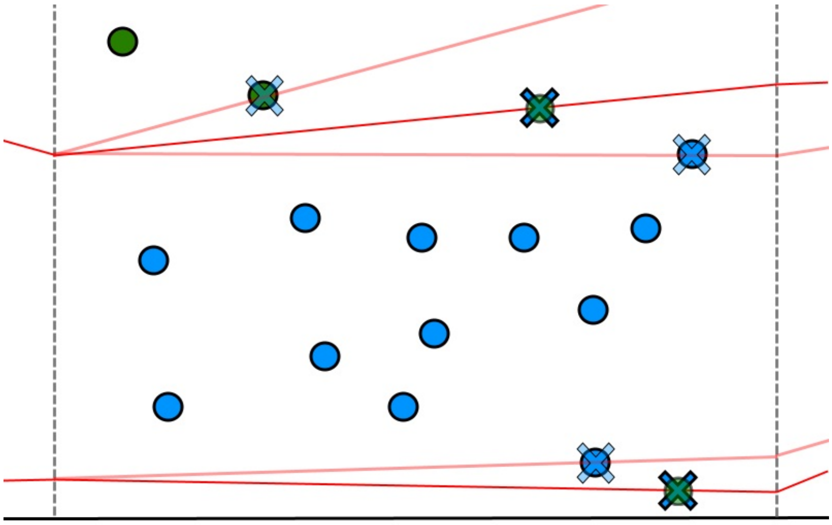

Figure 12: Possible operations for an iteration of the Simulated Annealing algorithm for an interval i

Where $\sigma(a, b)$ denotes the slope of point a to point b. Otherwise stated, $\mathcal{T}_i = (\mathcal{T}_i^{lb}, \mathcal{T}_i^{ub})$ is the set of points in interval i, which intersect with one of the bounds. Please note that $\mathcal{T}_i$ can only be empty when an interval contains no data points. In all other cases, taking a bound which intersects with no data point would never suffice the min/max-condition for $\hat{\sigma}$.

When one point needs to be added or removed from H^{in} and say the upper bound and υ_i are selected with p_i being the right end of the selected interval . Then, υ_i is increased or decreased until the first occasion that a point $\hat{\eta} = \theta \cdot \upsilon_i + (1 - \theta) \cdot \upsilon_{i-1}$. Then $\mathcal{T}_i^{ub} = \hat{\eta}$ and either this point is added to H^{in} or the previous tight point is removed from this set, depending on whether the objective was to add or to remove a point.

Otherwise stated, in the described scenario, if υ_i is selected, the directional coefficients σ of υ_{i-1} to all points in $H[i]$ are calculated. Define $\tilde{\sigma}$ as the ordered list σ and state that $\mathcal{T}_i^{ub}$ is the point t, with corresponding directional coefficient $\tilde{\sigma}_t$. Hence, the tight point was equal to the t'th element of this ordered list. Then, the new tight point is defined by $\mathcal{T}_i^{ub} = \tilde{\sigma}_{t+1}$ or $\mathcal{T}_i^{ub} = \tilde{\sigma}_{t-1}$, depending on whether an element needed to be added or subtracted from the set. Then also, $\upsilon_i' = \upsilon_i + \tilde{\sigma}_{t+1} \cdot (p_i - p_{i-1})$ or $\upsilon_i' = \upsilon_i + \tilde{\sigma}_{t-1} \cdot (p_i - p_{i-1})$. This is illustrated in figure 12.

When υ_i' is determined, all other values of υ preceding or succeeding i need to be adjusted to secure a piece-wise linear continuous function as a bound. This is done by starting at the new υ_i' and moving in the direction of the point which ensures the same size of H^{in} until the intersection of the next gridline. Every preceding or succeeding value of υ_j is replaced by υ_j', following this iterative procedure. This is illustrated for a random υ' in figure 13.

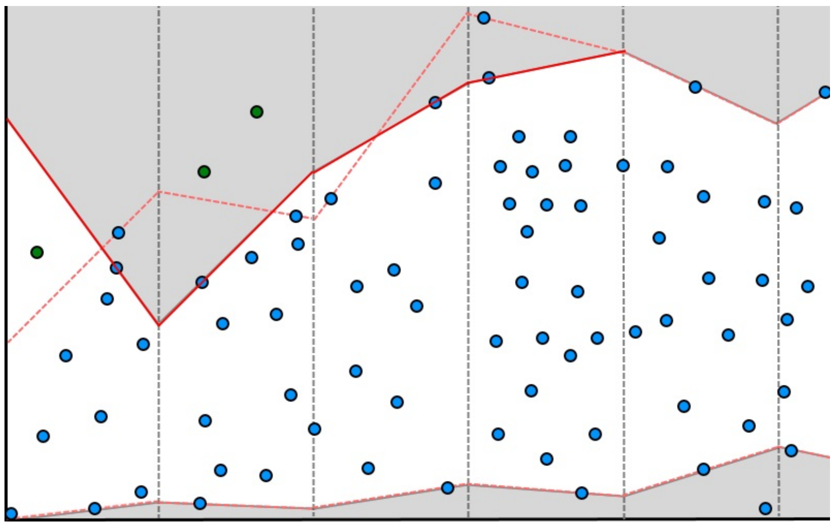

Figure 13: Altering of υ^{ub} after a point is removed from H^{in}

Step 2: Solution Quality

Note that in the initialization step a level α is chosen, which implies the number of points allowed to leave out by the algorithm. Otherwise stated, $|H^{in}| \geq (1-\alpha)\cdot|H|$. In practice, this is translated to a similar constraint on interval-level, i.e. $|H_i^{in}| \geq (1-\alpha)\cdot|H_i|\ \forall i$. However, the algorithm may propose solutions with $|H_i^{in}| < (1-\alpha)\cdot|H_i|$. This is to give the algorithm more freedom in the global search stages, and to decrease the probability of ending up in a local optimum due to insufficient search options with higher temperatures. These solutions do result in certain penalty costs equaling $\rho\cdot((1-\alpha)\cdot|H|-|H^{in}|)$. This implies that higher temperatures will allow for more of these bad solutions, where with lower temperatures the algorithm wants to increase the size of $|H^{in}|$ until $|H^{in}| \geq (1-\alpha)\cdot|H|$, given that the penalty costs are chosen large enough. Hence, the parameter ρ needs to be chosen such that the algorithm has the desired freedom in the first stages, whereas it detests these solution when the temperature decreases. Ideally, the parameter should also take into account the possibility that an extreme outlier should be left out, even if this means that $|H^{in}| \geq (1-\alpha)\cdot|H|$.

Step 3: Solution Costs
The calculation of the costs for the new solution υ' is defined as

$$\mathcal{C}(\upsilon') = \sum(\upsilon_{ub}-\upsilon'_{lb}) + \rho\cdot\sum_i \mathcal{P}[i]$$

where $\mathcal{P}[i] = (1-\alpha)\cdot|H_i| - |H_i^{in}|$ denote the penalty points of interval i. Hence, the costs are calculated similar to the initial costs with the addition of the penalty costs.

Step 4: Acceptance New Solution (Metropolis criterion)
The algorithm now needs to determine whether the newly proposed solution v' is accepted as the current solution. This happens directly if the costs of $\mathcal{C}(v') < \mathcal{C}(v)$, because we then have found a better solution than the predecessor according to our definition of the costs. When this is not the case, there new solution is accepted with a certain probability, as is the norm with SA.

This probability of acceptance is known as the Metropolis Criterion, and is defined as

$$e^{-\theta \cdot \Delta\mathcal{C}/T} > r$$

With $r \sim U(0,1)$, θ user-specified constant, $\Delta\mathcal{C}$ the difference in costs between v and v' and T the current value of the temperature parameter.

Thus, if this holds true the new solution is accepted, otherwise it is deflected. Observe that the probability that it holds true increases in T. This is what defines the main idea behind SA; the probability of accepting a worse random solution decreases as T decreases, resulting in a global search phase slowly evolving into a local search. Also note that the constant chosen which has a large impact on the performance of the algorithm, i.e. it co-determines the probabilities of accepting worse solutions for high and low values of T.

Step 5: Termination Iterative (Inner) Loop
Increase τ by one.
If $\tau < \bar{\tau}$: return to step 1
Else: go to step 6

Step 6: Termination Temperature (Outer) Loop
Decrease T by the specified fraction, i.e. $T \leftarrow \delta \cdot T$.
If $T > t$: reset τ to 0 and recalculate $\bar{\tau} = 100 + 20 \cdot T$. Return to step 1.
Else: algorithm terminates

Parameter Values
The values determined to be suited for adequate performance, without causing excessive running times of the algorithm, are presented in the next table. These parameters are determined by an extensive trial and error procedure.

Parameter	**Definition**	**Value**
T	Starting Temperature	10
t	Threshold	.01
δ	Cooldown rate	.9
ρ	Penalty Cost	$10 \cdot x_h$
n	Number of Gridlines	40
θ	Constant	$\frac{10}{x_h}$
α	Fraction of Points to Leave Out	.05

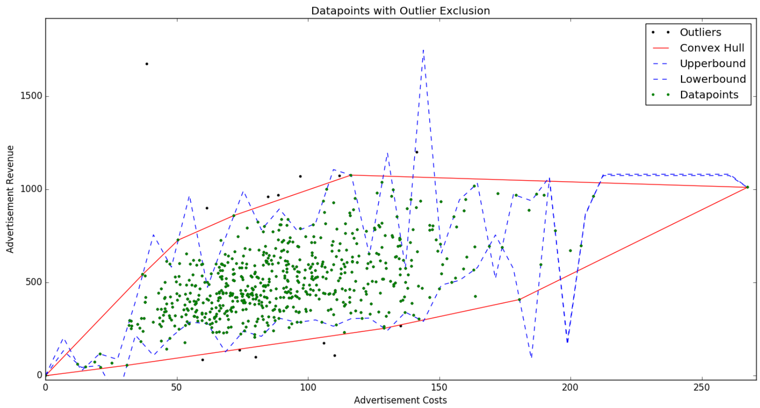

Figure 14: Conservative Solution for Outlier Exclusion

6.2.3 Conservative Solution

From the Simulated Annealing algorithm described, two piece-wise linear functions result, reflecting an upper and lower bound that determine which data points should be included in a $100 \cdot (1-\alpha)\%$ confidence interval and which do not. Due to the setup of the algorithm some of the data points are close to the interval bounds, giving them a bigger chance of being excluded of the set, since $\sigma(\upsilon_i, (y_k, x_k))$ heavily depends on the distance $|p_i - x_k|$. To correct for this inconvenience, we include a larger set than specifically depicted by the Simulated Annealing algorithm. This is done by locating the tight points $\mathcal{T}$ of υ^{ub} and υ^{lb}, i.e. the data points that intersect the line segments of these bounds, and take the convex hull of these points. This results in a set consisting of the data set resulting from the Simulated Annealing algorithm, with the addition of points sufficiently close to the nearest gridline, resulting in a disproportionate direction coefficient. Figure 14 represents this final solution and the data points ultimately left out.

6.3 Synthetic Data

In addition to the real world case from PauwR, artificial synthetic data was generated to test the validity of the model and to perform sensitivity analyses. This data consists of 4 distinct functions, which are known to have the desired monotonically non-decreasing concave shape. Moreover, they are based upon the recommendations of Bhattacharya (2010) and Bhattacharya (2012) when designing appropriate functions. The four created functions are:

1. $y = 5 \cdot \sqrt{x} + \epsilon$
2. $y = 10 \cdot \sqrt[3]{x} + \epsilon$
3. $y = .5 \cdot x + \epsilon$
4. $y = 10 \cdot \log x + \epsilon$

With $\epsilon_i \sim N(0,2)$ as the artificial error term. The final data points are created by first generating random x-values, with $x_i \sim U(0,100)$. The y-values per distribution follow directly from the definitions stated above. The generated data sets can be seen in the results section in figure 23-26.

The uncertainty sets are also generated artificially. The X-uncertainty, and the proportional part of the Y-uncertainty are set equal to (the average of) those previously defined for the PauwR case, i.e. 10% en 11% respectively. The fixed value uncertainty part of the Y-uncertainty is generated as $\bar{\beta}_k \sim UNIF(0,5)$, where $UNIF(\cdot)$ denotes the continuous uniform distribution and $\bar{\beta}_k$.

6.4 Random Generated Uncertainty

In order to test their average performances, the algorithms are tested on 50 randomly drawn samples of data points. These data sets are created by selecting random data points from their previously specified uncertainty intervals and boxes, respectively. This means that random values are drawn for the values of α_k and β_k for each data point $k \in K$. These values are individually drawn from an uniform distribution on the interval [-1,1], i.e. $\alpha_k \sim UNIF(-1,1)$ and $\beta_k \sim UNIF(-1,1)$. These drawings are performed without considering the budget of uncertainty Ω and the relative cost parameter γ, with the X-uncertainty magnitude $\bar{\alpha} = 0.1$ and the number of gridpoints set to 40.

7 Results

In this chapter, the performance and outcomes of the multiple regressions are presented. Firstly, the different regressions are performed on the artificially generated data. This is done for varying levels of the different parameters; the $\bar{\alpha}$, Ω, number of gridlines and γ. The different methods with varying parameter values are then compared based upon their respective R-squared values for the nominal data sets as well as those of their individual worst-case scenarios.

Secondly, the regression methods are implemented on the real-life data sets of PauwR. This is done with varying levels for Ω and γ, but for a fixed level of $\bar{\alpha} = 0.1$ and with the number of gridlines set to 40. The results for the varying parameter levels are mutually compared as well as with the results of the methods on the artificial data. Thirdly, the results of the regressions are given as input to the End-Model presented in Chapter 4.3 to solve the actual Budget Allocation Model and analyze the results. Finally, the actual need for and impact of the robust models is discussed in Chapter 7.3.

Please note that the figures illustrating the conclusions drawn in this chapter are presented in the chapter itself. However, the more extensive tables, which form the basis of these figures can be found in Appendix A. Also note that, in the figures where it states 'Average R-squared', the average is taken over the remaining parameters not equal to the dependent variable of the figure. The values of these parameters used to determine these averages are also given in the tables in Appendix A.

7.1 Artificial Data Results

7.1.1 Sensitivity Analysis X-Uncertainty

In Figure 15 the R-squared values of the nominal and robust solutions for differing values of the magnitude of uncertainty $\bar{\alpha}$ are depicted. A negative relation is seen between the $\bar{\alpha}$ and the worst-case values of the solutions. This is in line with expectations, because an increase in the magnitude of uncertainty is expected to worsen the worst-case scenario. However, the robust solution decreases more in the nominal case than it compensates in the worst-case with respect to the nominal solution. This indicates that, , when $\bar{\alpha}$ increases, it becomes hard to fit regressions that hedge against the worst-case scenario while maintaining a good performance in the nominal case. The shape of the curves also gives reason to believe that the solution value of the worst-case will decrease significantly if $\bar{\alpha}$ rises even further.

In Figure 16 the relation between the budget of uncertainty Ω and the average R-squared value of the nominal and robust solutions is presented. It shows that, if Ω rises, the solution value in the worst-case decreases. This is, again, in line with expectations due to trivial rationale. What does stand out here is that the marginal increments are not strictly decreasing. The point where this presumed relation fails is for $\Omega = 0.7$. The difference seems to be insignificant, so it could occur here due to measurement or rounding errors.

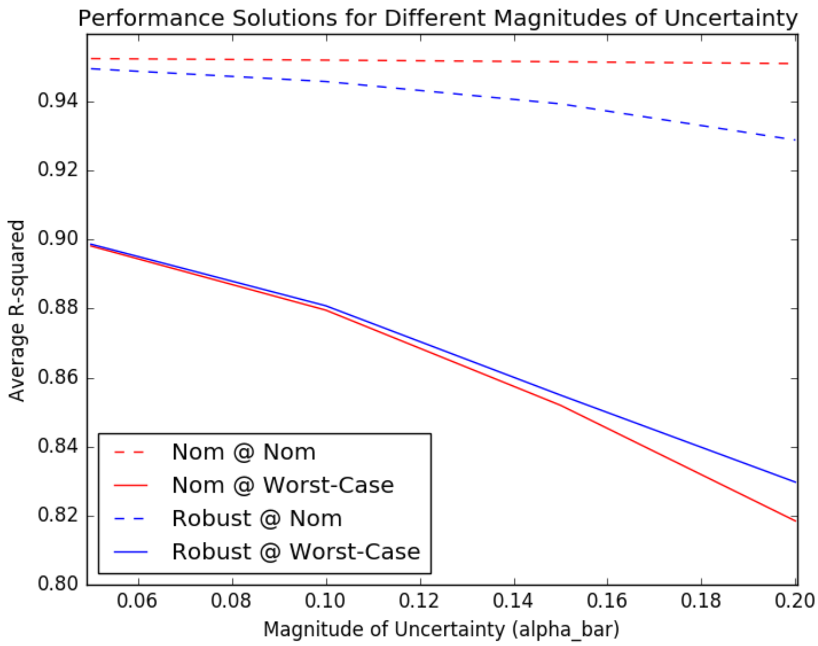

Figure 15: Comparison of Nominal and Robust Solutions with X-Uncertainty for different levels of $\bar{\alpha}$

In Figure 17 the effect of the number of gridpoints on the average R-squared values for the nominal and robust solutions is shown. We observe an expected monotonically non-decreasing shape with this effect. Moreover, when the number of gridpoints increases, the robust solution seems to be improving at a steeper rate than the nominal solution. This could be due to more flexibility to adjust for the datasets presented by the adversary when there are more gridlines. This is in contrast with the situation with few gridlines, where both solutions seem to be performing relatively poorly.

7.1.2 Sensitivity Analysis X- and Y-Uncertainty

In Figure 18 the nominal and robust solutions for different levels of γ are presented. A non-monotonical shape is observed here. Also, the robust solution seems to perform better for higher values of γ, i.e. when it is less costly to deviate with X-Uncertainty. Moreover, we observe a similar pattern for the quality of the robust solution as the pattern for the nominal solution. For both regressions the solution quality on average performs worse for low values of γ as opposed to high values of γ. This could be due to larger possible gains to be made with X-Uncertainty than with Y-Uncertainty for the Adversarial problem. This would follow from the way the uncertainty is setup in the artificial data. The X-Uncertainty is equal to 10%, whereas the Y-Uncertainty is on average

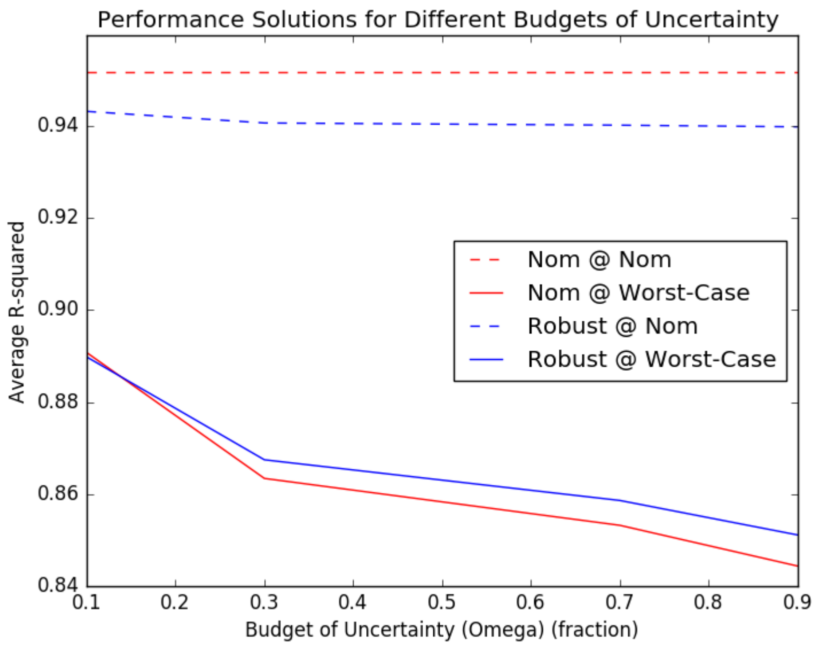

Figure 16: Comparison of Nominal and Robust Solutions with X-Uncertainty for different levels of Ω

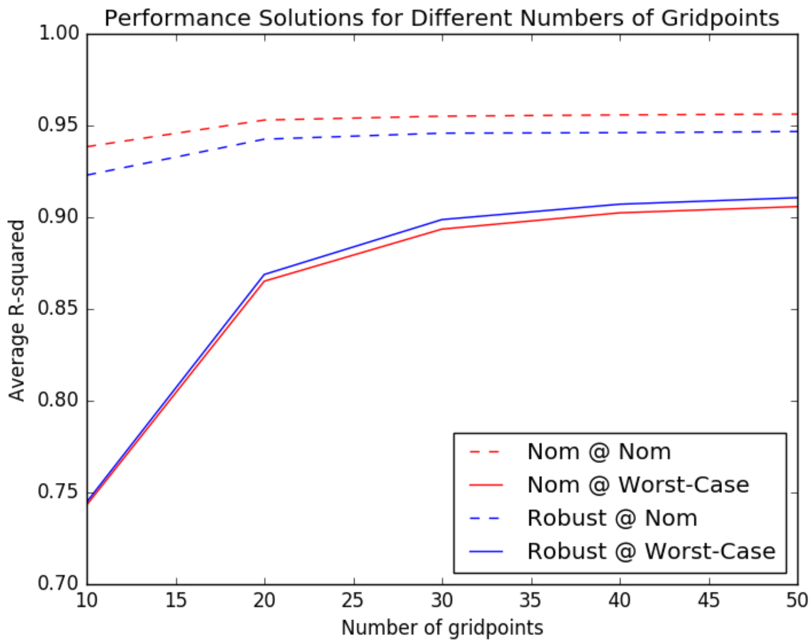

Figure 17: Comparison of Nominal and Robust Solutions with X-Uncertainty for differing numbers of gridlines

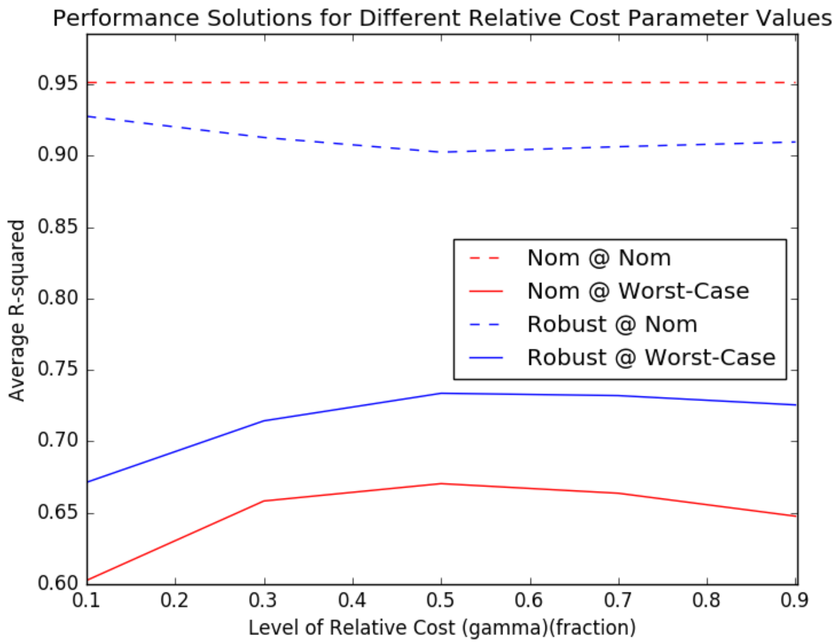

Figure 18: Comparison of Nominal and Robust Solutions with X- and Y-Uncertainty for different levels of γ

equal to approximately 15%. However, since the X-values range from 0 to 100 and the Y-values range from 0 to approximately 50, the absolute uncertainty of X is assumed to be higher. This could cause the adversary to be able to gain more from the X-Uncertainty.

In Figure 19 a similar graph is presented for differing levels of Ω with X- and Y-Uncertainty. The pattern shown is similar as with only the X-Uncertainty. The main difference is the steepness of the curve due to more possibilities with uncertainty, as well as a bigger budget of uncertainty in general due to the way it is setup in the Solution Approach. Recall that the budget of uncertainty is specified as a fraction of the total possible deviations of the data points. This implies that, in this setup, the set of total possible deviations has increased. As an example, for every point $k \in K$ to deviate maximally, one needed an $\Omega = 1$ with the X-Uncertainty case, where one only needs an $\Omega = \gamma$, with $\gamma \in (0, 1)$.

We also observe that the interval of the robust solution expands rapidly with increasing Ω, as well as with the nominal solution. However, the robust solution tends more towards the nominal solution when Ω increases. This could be due to the way the Y-uncertainty is setup. With X-Uncertainty, the data points are allowed to move horizontally, which gives a differing impact on the behavior of the robust solution for different nominal locations of the data points. As stated earlier, data points that lie below the regression line will tend to go to the right whereas data points above the line will tend to move to the left. In general, the tendencies for the Y-uncertainty are similar; points above the line will tend to go up where points underneath the line will tend to go down. However, there is a difference.

The difference follows from the difference in level of impact of data points that are close to the regression line and those that have a bigger distance from the line. Because Y-Uncertainty moves much more perpendicular to the regression line for higher values, in most of the cases the regression line needs to adjust more to account for these changes. Moreover, the data points for which the X-Uncertainty moves relatively perpendicular to the regression line, are for data points with lower y-values, and thus in general lower x-values. This means that, due to the level of uncertainty being relative to the value of x, the possible uncertainty added is smaller in general.

In conclusion, the Adversarial Approach needs to change its regression lines more over iterations in general when Y-uncertainty is applied. This causes more data points to switch side, underneath versus above, with respect to the regression line. The points that make this switch, in general have a greater impact with their Y-uncertainty than with their X-uncertainty. Thus, since these switching points have a greater influence, the regression is more focused towards these points and is willing to minimize their effect to a greater extend when more budget is allowed. This causes the algorithm to stay closer to the nominal values of these points, to minimize their effect, as does the nominal solution. Concluding that the best fit would tend more towards the nominal solution again.

In Figure 20 the number of gridpoints is the dependent variable. We recognize the same start of the shape in general as with the case of only X-Uncertainty. There are two main characteristics worth mentioning here. The first characteristic is that there is already a significant difference between the performance of the robust and the nominal solution for low numbers of gridpoints. The second characteristic of this figure that stands out, is the solution for 50 gridpoints performing slightly worse than with 40 gridpoints. In general this is counterintuitive. However this might be due to the placing of the gridpoints on the x-axis. Obviously, when gridlines are multiplied by an integer factor, a number of gridpoints is added, but the original set remains unchanged. When going from 40 to 50 gridpoints this is not the case. The gridpoints are evenly spread over the x-axis, resulting in every gridpoint being on a different position when going from 40 to 50 gridpoints. This could lead to less fortunate positions regarding the solution obtained. This raises the suspicion that there might be little to gain by adding more gridpoints from a certain threshold onward and that the improvements made in particular cases could be due to a fortunate gridpoint positioning.

7.1.3 Individual Channel Results

From Figure 21 we see that the averages of the robust solutions and the nominal SOCP solutions per channel are fairly equal in the case of X-Uncertainty. The solutions of the CHM appear to result in less accurate results on average, mostly due to a worse performance in the worst case. This could result from a relative

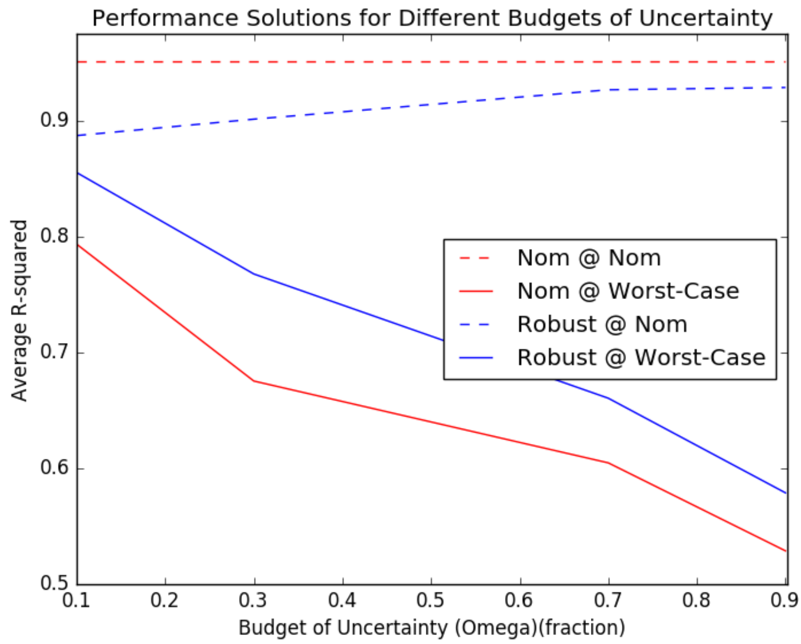

Figure 19: Comparison of Nominal and Robust Solutions with X- and Y-Uncertainty for different levels of Ω

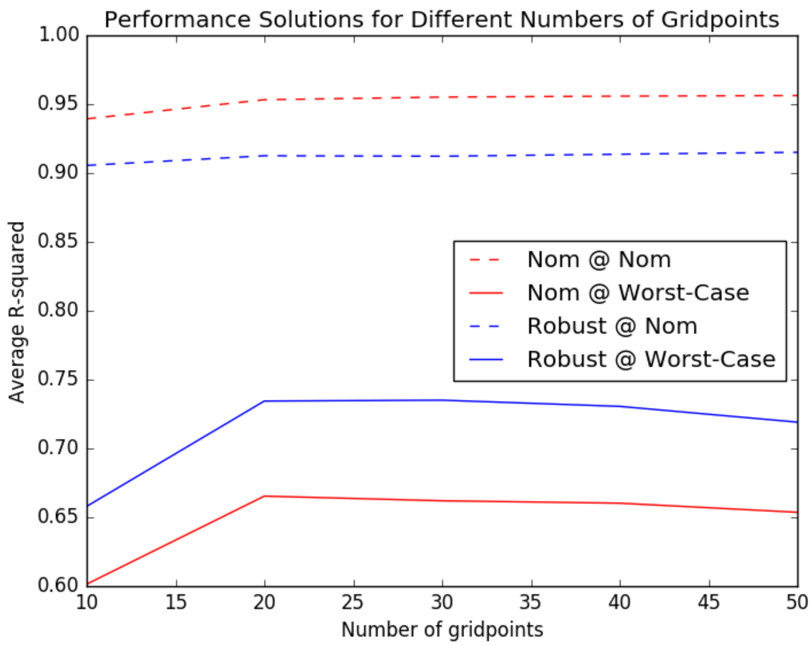

Figure 20: Comparison of Nominal and Robust Solutions with X- and Y-Uncertainty for differing numbers of gridlines

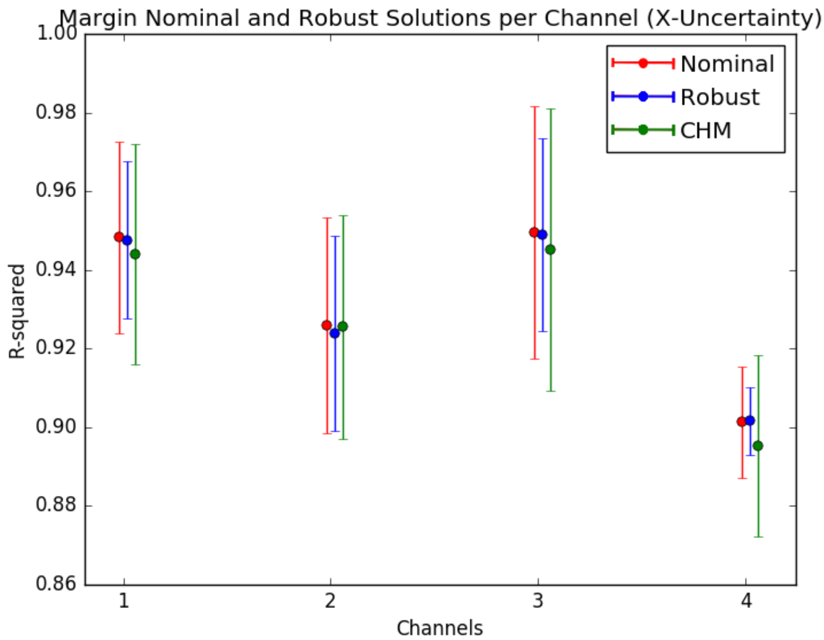

Figure 21: Performance Comparison of Nominal, Robust and CHM solutions for X-Uncertainty on the multiple artificially generated marketing channels

overfitting on the nominal data set, which still does not significantly outperform the nominal SOCP. It is also clear that the robust solutions give in on solution quality in the nominal case, but make up for it by having higher solution values in the worst-case, as is expected.

This figure also emphasizes the need for the data to take on the expected shape. For channel 3 we see a wider range of possible solutions, which is also the channel with the shape that deviates the most from the expectation, as can be seen in Figures 23-26. Even though the nominal solution is the best among the channels, the difference between the nominal and worst case is also the largest from the artificial data sets.

In Figure 22 we see a similar figure as the previous figure, now for the case with X- and Y-Uncertainty. With the added Y-Uncertainty, the robust solutions become more profitable. Moreover, the average of the robust solution is higher for all 4 channels, mainly due to a large profit in the worst-case compared to the nominal solutions. In addition, the intervals of margin are in general much larger as with X-Uncertainty only, as is expected.

Note that CHM is omitted from this figure due to its poor performance in the worst case with X-Uncertainty already. This makes it already clear that the solution would not be worth considering with even more uncertainty.

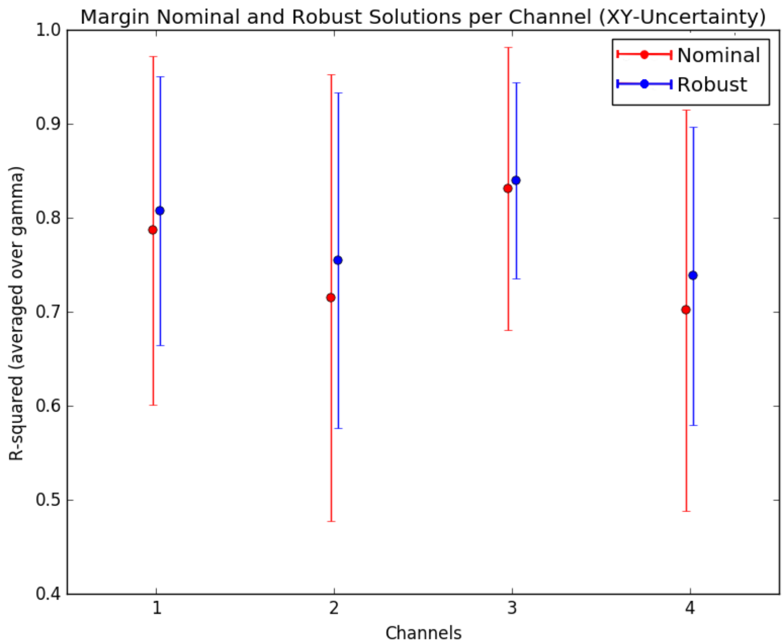

Figure 22: Performance Comparison of Nominal and Robust solutions for X- and Y-Uncertainty on the multiple artificially generated marketing channels

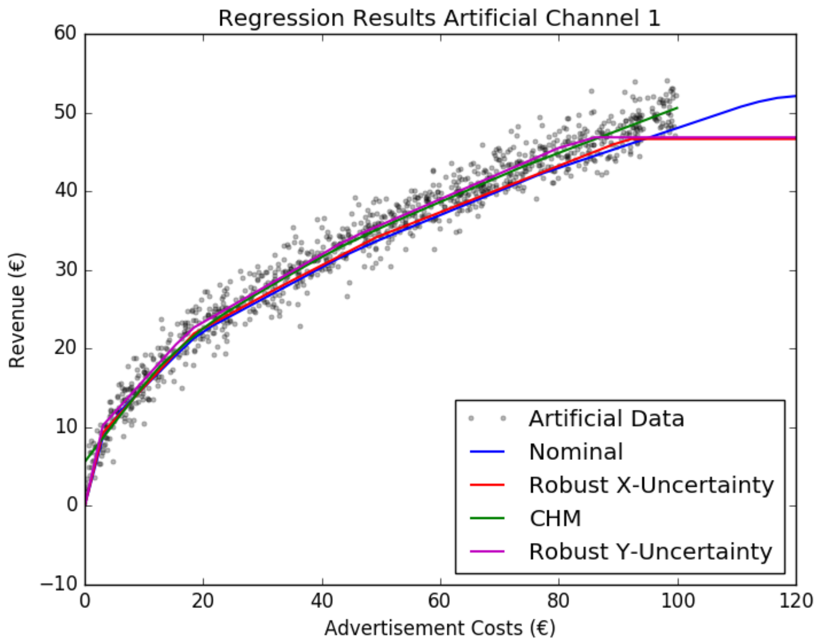

Figure 23: Fitted Regressions for artificially generated data channel 1

In Figures 23-26 we see the regression lines plotted for all four artificial data sets. We observe that the robust solutions flatten at the end, due to the larger allowed deviations of the data points near the right end of the x-axis, because of the proportional uncertainty. This causes the robust lines to flatten early to hedge against the huge costs of single points above the line moving left or points underneath the line moving right. In general this leads to the robust solutions having a greater arc. Otherwise stated, at the start the robust solutions increase quicker, but flatten out sooner as well. Also, in general, the CHM gives a bigger arc than the nominal solution as well. This is presumably due to the convex hull step of the algorithm. Roughly, the shapes seem to fit the data fairly well.

7.2 Real Data Results

From the figures and tables presented in the upcoming sections, it is immediately clear that the general results for the real data are not nearly as good as with the artificial data. On average, the R-squared is about 0.1. This suggests that there is plenty of room for improvement. However, there are also some analyses to be done on the mutual regressions performed and the current results.

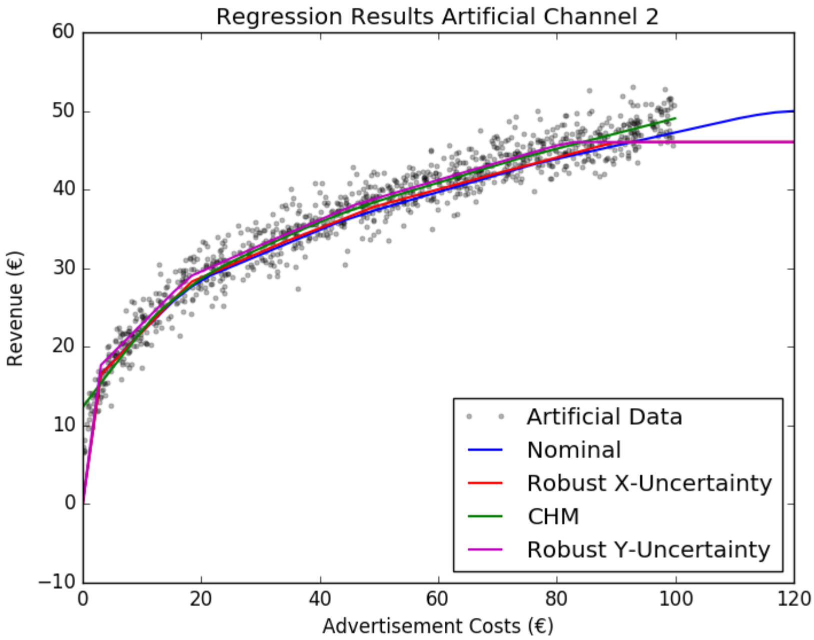

Figure 24: Fitted Regressions for artificially generated data channel 2

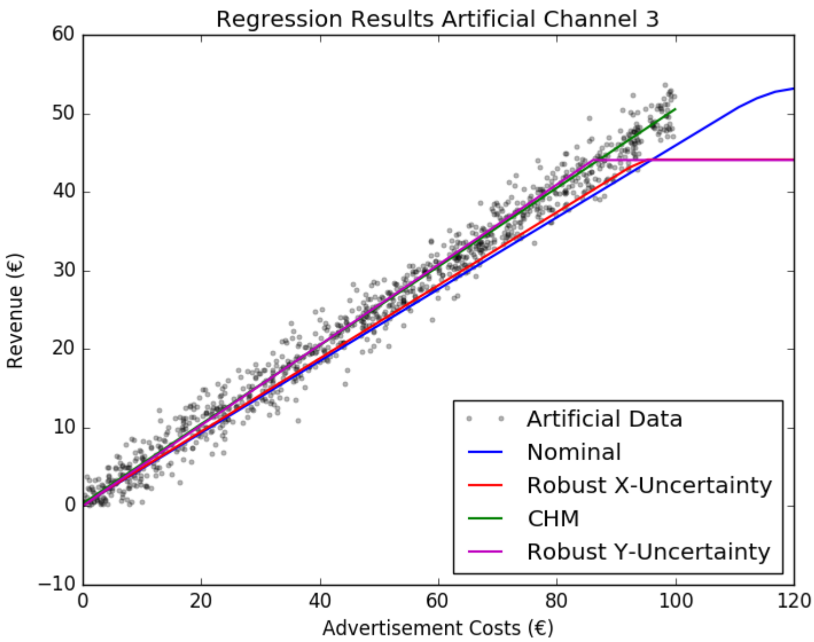

Figure 25: Fitted Regressions for artificially generated data channel 3

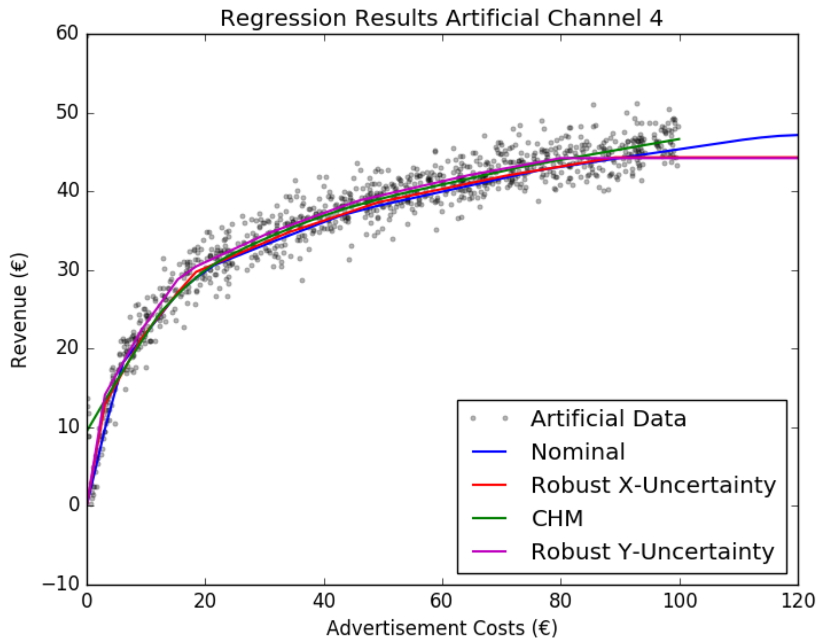

Figure 26: Fitted Regressions for artificially generated data channel 4

7.2.1 Outlier Exclusion

As was already stated in Section 5.5, the real data sets are almost certain to have some outliers. Hence, in order to nullify the impact of such outliers in the final results, the Simulated Annealing algorithm was used to exclude these outliers. The parameters for the algorithm were equal to the values given already in Section 5.5, giving the following results:

Campaign	**Initial Size**	**Post-SA Size**	**Initial** R^2	**Post-SA** R^2
1	647	641	0.10	0.12
2	647	636	0.13	0.17
3	647	644	0.08	0.10
4	647	643	0.12	0.12
5	647	633	0.09	0.14
6	647	638	0.08	0.10
7	647	634	0.33	0.40
8	647	640	0,18	0.21
9	647	634	0.20	0.22
10	647	638	0.09	0.10

Where the sizes denote the sizes of the data sets before the SA algorithm was performed and afterwards, and the R^2 denotes the R-Squared values corresponding to the nominal regressions on the nominal data sets before and after the SA algorithm was performed on the data sets.

The first remark that can be made regarding the table is that, even though the algorithm was allowed to leave out approximately 5% of the data points, it never left out more than 3.7%, and on average only leaves out 1.4%. This indicates that the conservative final step does lead to significantly more conservative exclusions.

The second remark is that this table seems to indicate that these outliers do have a significant impact on the regression performance. Even by only leaving out 1.4% of the data points on average, the R-squared value increases by 0.026 on average, which is equal to an average change of 15.5%. This seems to indicate that the outliers are of a severe level, and that their values differ extremely from the rest of the data set. This gives reason to believe that outlier exclusion is the right decision for these data sets.

7.2.2 Regression Method Comparison

In Figure 27 it is immediately obvious that overall performance of the algorithms is much worse than with the artificial data. This is due to the wider scattering of the data points and a less obvious shape fitting the predetermined general shape of the regressions. What also stands out is the terrible performance of the CHM for campaigns 1, 3, 4 and 10. This is presumably due to the large distortions of the CHM following non-obvious concave or convex shapes of the data points. The LOESS regressions performed in the algorithm lead to such a fluctuating estimation that the algorithm fails to perform accurately. For the other campaigns the CHM performs consistently poorer than the SOCP formulations.

With the real data the averages of the robust solutions seem to be slightly better than the averages of the nominal solutions for the X-Uncertainty. Moreover, the robust intervals are significantly smaller, as is expected.

With X- and Y-Uncertainty in Figure 22 we see similar patterns of the performances as with only X-Uncertainty. However, the magnitude of the intervals is significantly increased, implying less certainty. This is in line with expectations for trivial reasons. Again, on average, the performance of the robust solutions are approximately equal to the performance of the nominal solutions. However, the robust solutions provide smaller intervals of variance regarding the solution quality, making it the preferred method with risk-averse clients.

7.2.3 End-Model Results

In Figure 29 the results are presented when the different solutions are given as input to the final budget allocation model as presented in Chapter 4.3. A few findings are done for these results. Firstly, the CHM is a more optimistic estimate compared to the robust estimations, i.e. the expected payoff of an investment is higher according to the CHM as opposed to the robust solutions. The CHM estimates start very steep, but flattens out quicker than the nominal solution. Secondly, for the cases where the relative costs parameter $\gamma = 0.5$,

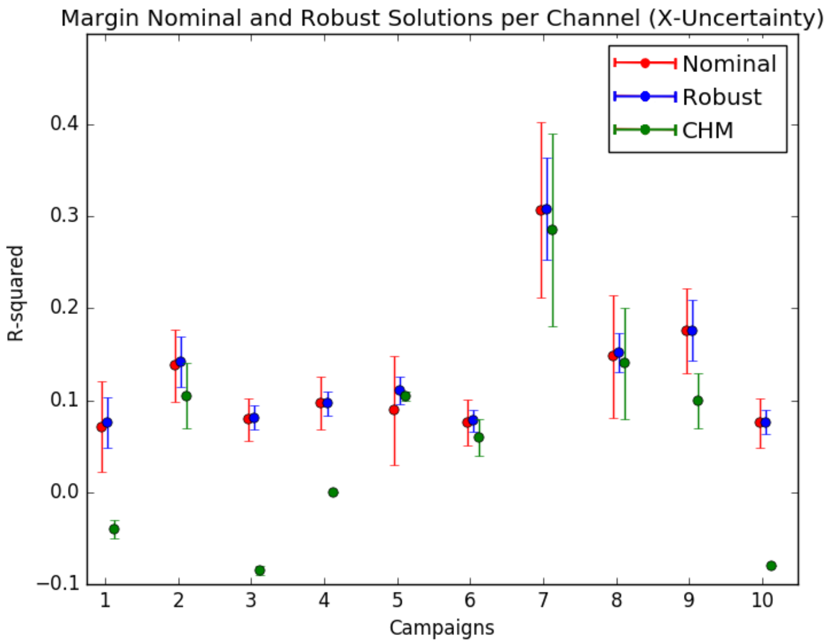

Figure 27: Performance Comparison of Nominal, Robust and CHM solutions for X-Uncertainty on the multiple real-life marketing campaigns

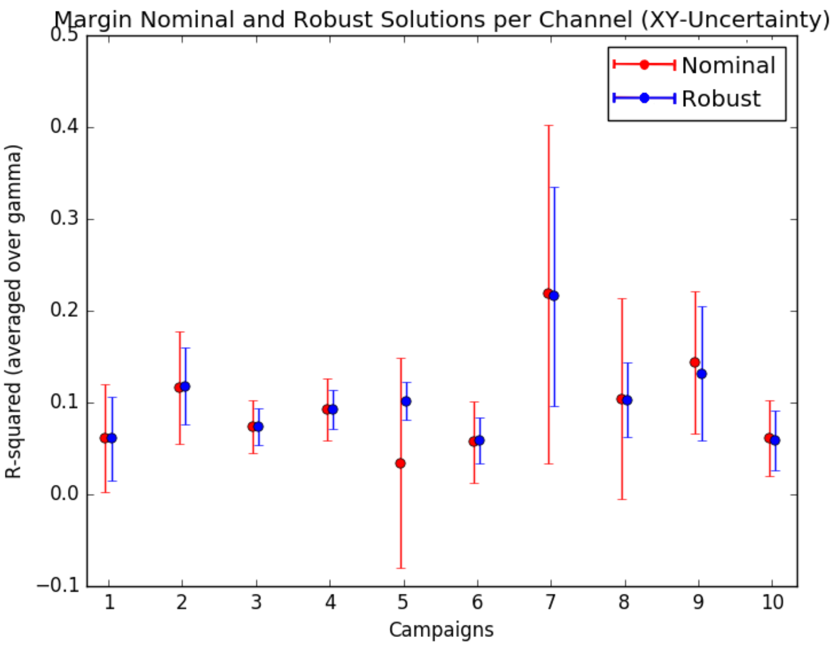

Figure 28: Performance Comparison of Nominal and Robust solutions for X- and Y-Uncertainty on the multiple real-life marketing campaigns

the revenue curve starts relatively low, but ends high compared to the curves with other values of γ. Thirdly, the nominal case gives the lowest revenues for lower budgets, but gives the highest revenues when the budget increases. This could be a result of the flattening that happens for higher cost values with the robust regression methods, as discussed earlier. The CHM presumably flattens more for the higher cost values due to the convex hull step in this algorithm.

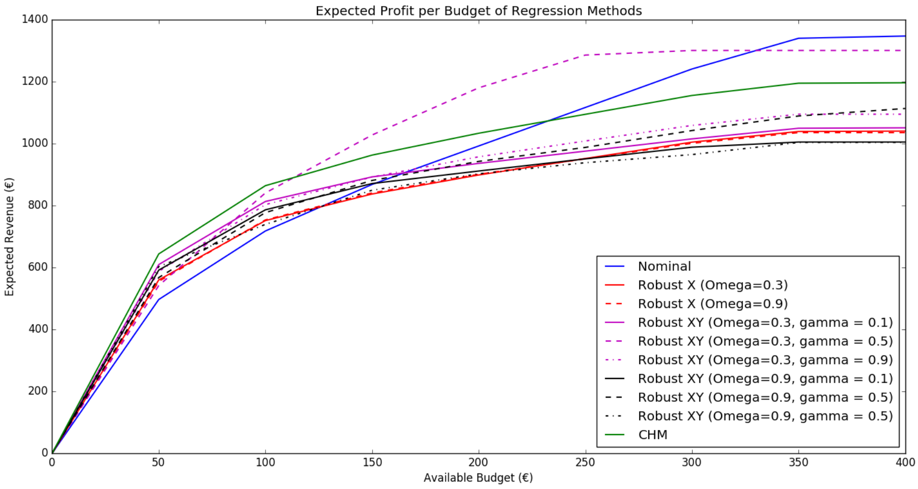

Figure 29: Results of End Model for different Regression Methods with varying budgets available

7.3 Need for Robust Solutions

The results in this chapter mainly show the comparisons between the nominal and the robust solutions. When comparing the nominal-case with the worst-case solutions for the artificial data, as is seen in Figure 21 and 22, the robust solutions do generate smaller intervals. In the case where there is only uncertainty in the costs, the averages of the performance between the nominal and robust solution is approximately equal. The need for a robust solution seems questionable. The nominal solution performs slightly better in the nominal case, where it performs slightly worse in the worst-case. Overall, the results are not conclusive whether the robust solution actually outperforms the nominal solution when looking at extreme cases.

With uncertainty in both the costs and revenues, the robust solution does seem to outperform the nominal solution with the artificial data on the extreme cases. The average of the robust solution is better for each channel, as is seen in Figure 22. Moreover, the robust solution exceeds the nominal solution in the worst-case tremendously. Hence, with all uncertainty included, the artificial data favors the robust solution.

With the real-life data similar plots are given in Figures 27 and 28. With only X-uncertainty, the solutions behave similarly to the artificial data, relatively equal to each other. So, the robust solution produces slightly smaller intervals, but the averages do not differ significantly. With X- and Y-uncertainty, the differences between the robust and nominal solutions are less significant as with the artificial data. The robust solutions do however prevent some disastrous outcomes with certain campaigns. This does favor the robust solutions, especially for risk-averse clients.

To get more insight in the performances of both the nominal as the robust solutions, the random data is generated in accordance with the description in Chapter 6.3. This resulted in Figures 30-33. Thus, we get the R-squared values of the robust and nominal solutions in 50 randomly generated possible scenarios from the uncertainty sets previously specified. These scenarios are generated by selecting points at random from every uncertainty interval and box for the case with X-Uncertainty and with X- and Y-Uncertainty respectively. So, these cases could be viewed as, with the limited information available about the distribution of the uncertainty, average case scenarios of the dataset. Hence, these R-Squared values give an indication of how the algorithms perform when considering more conservative situations as opposed to the possible extreme cases of the data.

From Figures 30 and 31 it is readily seen that the nominal solutions outperform the robust solutions for all 4 campaigns. This can be concluded because the averages of the nominal solutions are higher than those of the robust solutions. Moreover, because both the highest and lowest values of the nominal solutions are better than the highest and lowest values of the robust solutions respectively. Particularly for channels 2 and 4, the robust solution is far inferior to the nominal solution. In the case with both uncertainties as opposed to only X-Uncertainty, the robust solutions win some ground, but the nominal solutions stay superior. It is also immediately clear that with averaged uncertain data, when randomly generated as stated earlier, no data set has even approached the worst-case scenario. This is also the part where the nominal solutions gain terrain on the robust solutions.

From Figures 32 and 33 we may conclude that the same patterns are seen with the real-life data sets. Now for campaigns 1 and 5, we do see that the robust solution actually performs better than the nominal solution. This follows from the same reasoning as with the artificial data. More specifically, by looking at the distribution of the obtained R-squared values of the nominal and robust solutions, with the main focus on the maxima, minima and averages. However, these differences do not seem to be very large, especially when compared to the differences in solution quality on the remaining channels, clearly favoring the nominal solutions.

In conclusion, the need for robust solutions is debatable. When considering the extreme cases, the robust solutions seem to be outperforming the nominal solutions on average, particularly with the artificially generated data. However, when considering the random generated uncertainties the nominal solutions are significantly superior. With the real-life data, the same patterns are observed.

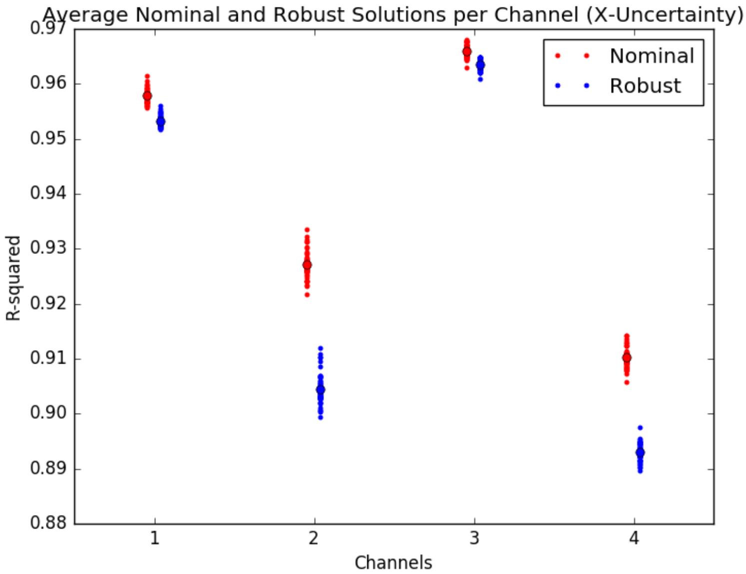

Figure 30: Average Performance Comparison of Nominal and Robust solutions for X-Uncertainty on the artificially generated marketing channels. Please note that the larger dots are the averages.

Furthermore, the overall low R-squared values for the real-life data make it even more doubtful. It is questionable whether one would cut on the nominal performance in order to hedge against the worst-case when the nominal R-squared values are this low already.

Hence, the results presented only significantly favor the robust solutions with the artificial data and uncertainty in both the costs and revenues. Furthermore they do not significantly outperform the nominal solutions and are even inferior to the nominal solutions in most cases considered. In conclusion, more research is needed to determine what realistic expectations of the uncertainties are, to definitively state whether robust solutions are actually desired.

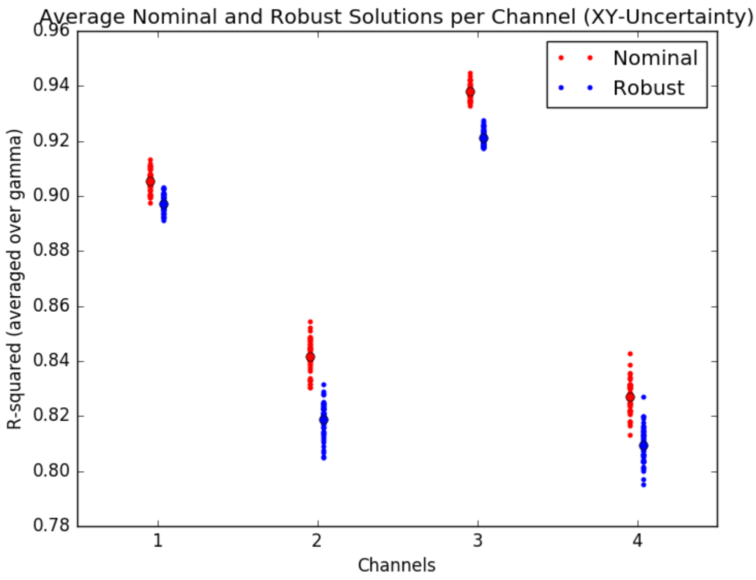

Figure 31: Average Performance Comparison of Nominal and Robust solutions for X- and Y-Uncertainty on the artificially generated marketing channels. Please note that the larger dots are the averages.

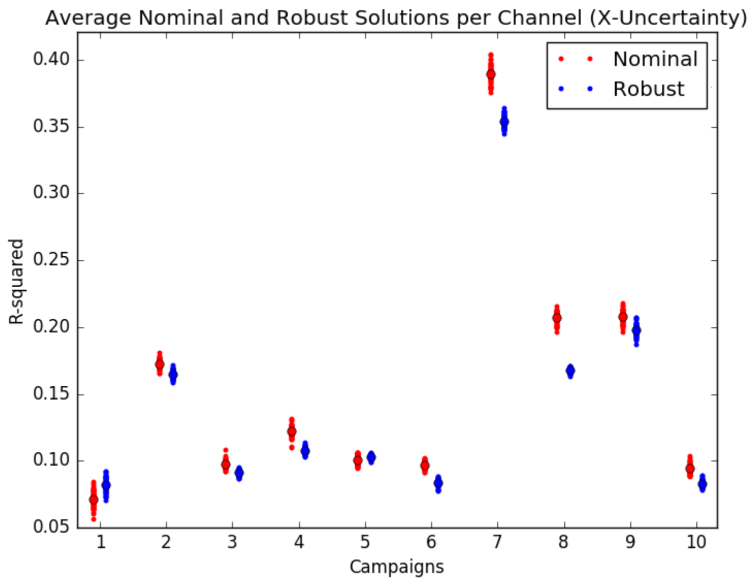

Figure 32: Average Performance Comparison of Nominal and Robust solutions for X-Uncertainty on the real-life marketing campaigns. Please note that the larger dots are the averages.

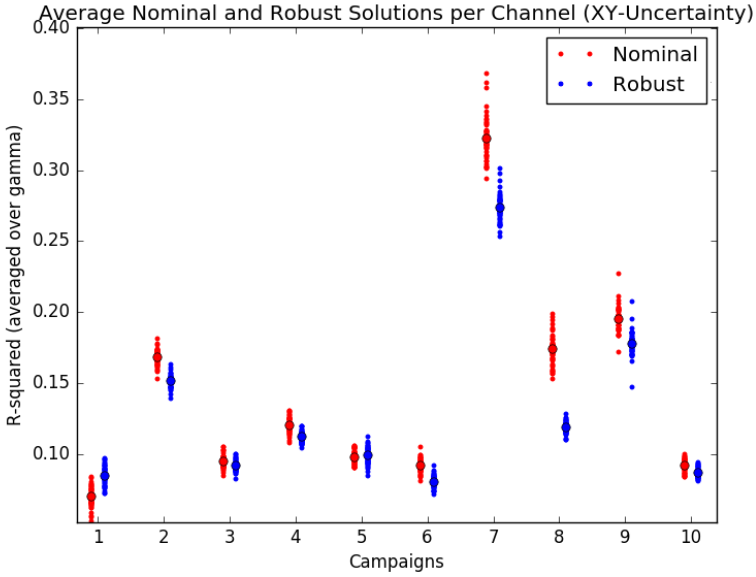

Figure 33: Average Performance Comparison of Nominal and Robust solutions for X- and Y-Uncertainty on the real-life marketing campaigns. Please note that the larger dots are the averages.

8 Conclusions & Discussions

In this thesis a mathematical model is proposed to solve the budget allocation problem among multiple marketing channels. The setting used is AdWords campaigns, the paid-link advertisement platform of Google Inc. It is a relatively simple model, solely based on history data, with the challenge being the modeling of the monotonically non-decreasing concave objective function.

The objective function is determined in a deterministic setting via the so-called Convex Hull Method, and a Second-Order Cone Problem formulation. In addition, robust models are proposed based upon the SOCP formulation, with uncertainty in the advertisement costs and revenues of the history data. The performance of the Convex Hull Method was far inferior to the SOCP for both the nominal cases as well as the worst-case scenarios. In addition, the results ultimately remained inconclusive regarding the desirability for the robust solutions in general.

The performance of the regression models is tested on multiple artificially generated data sets with varying parameters. With an average R-squared of approximately 0.8-0.9 for the specified values of the parameters. This leads to satisfying results from a practical point of view. In contrast, the performance on real-life data from a client of PauwR is relatively poor. The R-squared values for this data ranges from -.08 in the extreme case with a very large level of uncertainty to .42 in the best case, with an average of .1. These values indicate that large improvements of the proposed model can be made, based upon better estimations of the sales function. Presumably, better results can be obtained by identifying factors that influence the revenue generation per marketing campaign other than solely the budget invested.

A second direct improvement of the model could follow from better available data regarding inter-channel relations, carry-over effects of the marketing campaigns or the actual expenditure of Google Inc. in the history data. Extra data on these subjects can directly lower the level of uncertainty, and hence produce better estimates.

Another possible improvement of the model is to establish a greater understanding of the nature of the factors of uncertainty, and/or to model parts of the uncertainty directly in the end-model. This takes away some of the assumptions made in the current model and could lead to more accurate results.

The last desired improvement stated in this thesis is to find a accurate description for the uncertainties. Then the problem can be solved to optimality and take away the need for a heuristic, which resulted in approximations. Nevertheless, the solutions in this thesis are found via the Adversarial Approach, which is known to produce good approximations, leading to high quality solutions.

More research is necessary to draw definitive conclusions and implement the model in real-life.

References

[1] Aguilera, N. and Forzani, L. and Morin, P. (June 2010). On Convex Regression Estimators. Statistics Theory.

[2] Alizadeh, F. and Goldfarb, D. (2004). Second-order cone programming. Mathematical Programming, Volume 95, Issue 1, p.3-51.

[3] Anderson, E.D. (February 2016). On formulating quadratic functions in optimization models. MOSEK: Technical Report.

[4] Bertsimas, D. and Gupta, V. and Kallus,N. (August 2013). Data-Driven Robust Optimization. Operations Research.

[5] Bertsimas, D. and Thiele, A. (March 2006). Robust and Data-Driven Optimization: Modern Decision-Making Under Uncertainty. INFORMS, p.95-122.

[6] Bertsimas, D. and Sim, M. (2004). The Price of Robustness. INFORMS, Volume 51, Issue 1, p.35-53.

[7] Bhattacharya, P.(2012). GAM in Marketing Mix Modeling: Revisited. Jigyasa Analytics.

[8] Bhattacharya, P.(2008). Marketing Mix Modeling: Techniques and Challenges. NESUG.

[9] Bhattacharya, P.(2009). Optimizing the Marketing Mix. NESUG.

[10] Bhattacharya, P.(2010) Using Generalized Additive Models in Marketing Mix Modeling. NESUG.

[11] Bienstock, D. and Ozbay, N.(November 2006) Computing Robust Basestock Levels. Elsevier, Volume 5, Issue 2, p.389-414.

[12] Boyd, S. and Vandenberghe, L. (2004) Convex Optimization. Cambridge University Press.

[13] Choi,C. and Lee, J.J. (1998). Chaotic Local Search Algorithm. Artificial Life and Robotics, Vol. 2, Issue 1, p.41-47.

[14] Cleveland, W.S. (December 1979). Robust Locally Weighted Regression and Smoothing Scatterplots. Journal of the American Statistical Association, Volume 74, Issue 368, p.829-836.

[15] Corana, A. et al. (September 1987). Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm. ACM: Transactions on Mathematical Software, Vol. 13, Issue 3, p.262-280.

[16] Giagkoulas, V. (2011) Optimizing the Marketing Budget Allocation Problem using Langrangean Based Techniques. Erasmus University Rotterdam.

[17] Hawkins, D.M. (1980). Identification of Outliers. Chapman and Hall,London.

[18] Goh, J. and Sim, M. (2010) Distributionally Robust Optimization and Its Tractable Approximations. INFORMS, Vol. 58, Issue 4, Part 1, p.902-917.

[19] Gorissen, B.L. and Yanikoglu, I. and Den Hertog, D. (January 2015). A Practical Guide to Robust Optimization. math.OC, Vol. 53, p.124-137.

[20] http(1). https://www.cmocouncil.org/facts_stats (2015). Marketing Facts and Stats. CMO Council, December 15, 2015.

[21] http(2). http://thelawdictionary.org/marketing-channel/ (2016) What is a Marketing Channel? The Law Dictionary, May 16,2016.

[22] http(3). https://support.google.com/adwords/answer/6304?hl=en (2015) Campaign: Definition. Google Adwords Support, December 15, 2015.

[23] http(4). http://www.businessdictionary.com/definition/marketing- budget.html(2016). What is Marketing Budget? Online Business Dictionary, January 12, 2016.

[24] http(5). https://support.google.com/adwords/answer/1722122 (2016). Understanding Ad Position and Ad Rank. Google Adwords Support, May 16, 2016.

[25] http(6). https://support.google.com/adwords/answer/2375454?hl=en (2016). Choose Your Bid and Budget. Google Adwords Support, May 16, 2016.

[26] http(7). https://support.google.com/adwords/answer/1704424?hl=en (2016). How Costs Are Calculated in AdWords. Google Adwords Support, May 16, 2016.

[27] Jacoby, W.G. (2000). Loess: a nonparametric, graphical tool for depicting relationships between variables. Pergamon, Electoral Studies 19 (2000), p. 577-613.

[28] Lancaster, K.J. (1966). A New Approach to Consumer Theory. Journal of Political Economy, Vol. 74, No. 2, p. 132-157.

[29] Lobo et al. (1998). Applications of Second-Order Cone Programming. Linear Algebra and its Applications, Vol. 284, Issues 1-3, p. 193-228.

[30] Lutenegger, A.J. (1979). Random-walk Variable Wind Model for LOESS Deposits. Retrospective Theses and Dissertations, Paper 7294.

[31] Miyashiro, R. and Takano, Y. (2013). Mixed Integer Second-Order Cone Programming Formulations for Variable Selection. European Journal of Operational Research, Vol. 247, Issue 3, p. 721-731.

[32] Moore, D.S. and McCabe, G.P. (1999). Introduction to the Practice of Statistics. W.H. Freeman Co Ltd.

[33] Naik, P.A. and Raman, K. and Winer, R.S. (2005). Planning Marketing-Mix Strategies in the Presence of Interaction Effects. Marketing Science, Vol. 24, No. 1, p. 25-34.

[34] Osborne, J.W. and Overbay, A. (2004). The power of outliers (and why researchers should always check for them). Practical Assessment, Research and Evaluation.

[35] Sim, M. (May 2004). Robust Optimization. Submitted to the Sloan School of Management.

[36] I. Yanikoglu, I. (2014). Robust optimization methods for chance constrained, simulation-based, and bilevel problems. Tilburg: CentER, Center for Economic Research

[37] Yanikoglu, I. and Den Hertog, D. (2011). Safe Approximations of Chance Constraints Using Historical Data. INFORMS, Vol. 25, Issue 4, p. 666-681.

A Tables

Please note that in this appendix the abbreviation gr denotes gridlines and the abbreviation B is used for Budget. Also note that Ω, $\bar{\alpha}$ and γ have the same definitions as throughout this thesis, i.e. the budget of uncertainty, level of X-Uncertainty and the relative costs of α and β respectively.

A.1 Artificial Data

A.1.1 X-Uncertainty Nominal R-Squared Values

Ω	$gr\backslash\bar{\alpha}$	0.05	0.1	0.15	0.2
0.1	10	(0.94,0.77)	(0.93,0.76)	(0.93,0.73)	(0.93,0.69)
	20	(0.95,0.90)	(0.95,0.88)	(0.95,0.85)	(0.95,0.81)
	30	(0.95,0.92)	(0.95,0.91)	(0.95,0.88)	(0.95,0.85)
	40	(0.95,0.93)	(0.95,0.91)	(0.95,0.89)	(0.95,0.86)
	50	(0.95,0.93)	(0.95,0.92)	(0.95,0.89)	(0.95,0.86)
0.3	10	(0.94,0.77)	(0.93,0.75)	(0.93,0.73)	(0.93,0.67)
	20	(0.95,0.90)	(0.95,0.87)	(0.95,0.83)	(0.95,0.80)
	30	(0.95,0.92)	(0.95,0.90)	(0.95,0.87)	(0.95,0.83)
	40	(0.95,0.93)	(0.95,0.91)	(0.95,0.88)	(0.95,0.84)
	50	(0.95,0.93)	(0.95,0.91)	(0.95,0.88)	(0.95,0.84)
0.5	10	(0.94,0.76)	(0.93,0.74)	(0.93,0.71)	(0.93,0.63)
	20	(0.95,0.89)	(0.95,0.86)	(0.95,0.81)	(0.95,0.76)
	30	(0.95,0.92)	(0.95,0.89)	(0.95,0.84)	(0.95,0.79)
	40	(0.95,0.92)	(0.95,0.89)	(0.95,0.85)	(0.95,0.80)
	50	(0.95,0.93)	(0.95,0.90)	(0.95,0.85)	(0.95,0.80)
0.7	10	(0.94,0.76)	(0.93,0.73)	(0.93,0.70)	(0.93,0.63)
	20	(0.95,0.89)	(0.95,0.85)	(0.95,0.80)	(0.95,0.74)
	30	(0.95,0.91)	(0.95,0.88)	(0.95,0.83)	(0.95,0.78)
	40	(0.95,0.92)	(0.95,0.89)	(0.95,0.84)	(0.95,0.79)
	50	(0.95,0.92)	(0.95,0.89)	(0.95,0.85)	(0.95,0.79)
0.9	10	(0.94,0.76)	(0.93,0.73)	(0.93,0.69)	(0.93,0.61)
	20	(0.95,0.89)	(0.95,0.85)	(0.95,0.80)	(0.95,0.74)
	30	(0.95,0.91)	(0.95,0.88)	(0.95,0.82)	(0.95,0.77)
	40	(0.95,0.92)	(0.95,0.88)	(0.95,0.84)	(0.95,0.78)
	50	(0.95,0.92)	(0.95,0.89)	(0.95,0.84)	(0.95,0.79)

A.1.2 X-Uncertainty Robust R-Squared Values

Ω	$gr\backslash\bar{\alpha}$	0.05	0.1	0.15	0.2
0.1	10	(0.93,0.77)	(0.93,0.76)	(0.92,0.73)	(0.90,0.71)
	20	(0.95,0.90)	(0.94,0.88)	(0.94,0.85)	(0.93,0.82)
	30	(0.95,0.93)	(0.94,0.91)	(0.94,0.88)	(0.93,0.86)
	40	(0.95,0.93)	(0.95,0.92)	(0.94,0.89)	(0.93,0.87)
	50	(0.95,0.94)	(0.95,0.92)	(0.94,0.90)	(0.93,0.88)
0.3	10	(0.93,0.77)	(0.93,0.75)	(0.92,0.72)	(0.89,0.68)
	20	(0.95,0.90)	(0.94,0.87)	(0.94,0.84)	(0.93,0.81)
	30	(0.95,0.92)	(0.94,0.90)	(0.94,0.87)	(0.93,0.85)
	40	(0.95,0.93)	(0.94,0.91)	(0.94,0.88)	(0.93,0.86)
	50	(0.95,0.93)	(0.95,0.91)	(0.94,0.89)	(0.93,0.86)
0.5	10	(0.93,0.76)	(0.92,0.74)	(0.92,0.70)	(0.89,0.65)
	20	(0.95,0.89)	(0.94,0.86)	(0.94,0.82)	(0.92,0.78)
	30	(0.95,0.92)	(0.94,0.89)	(0.94,0.85)	(0.93,0.82)
	40	(0.95,0.93)	(0.94,0.90)	(0.94,0.86)	(0.93,0.83)
	50	(0.95,0.93)	(0.95,0.90)	(0.94,0.87)	(0.93,0.83)
0.7	10	(0.93,0.77)	(0.93,0.74)	(0.92,0.70)	(0.89,0.65)
	20	(0.95,0.89)	(0.94,0.86)	(0.94,0.82)	(0.92,0.77)
	30	(0.95,0.92)	(0.94,0.89)	(0.94,0.85)	(0.92,0.81)
	40	(0.95,0.92)	(0.94,0.90)	(0.94,0.86)	(0.93,0.82)
	50	(0.95,0.93)	(0.95,0.90)	(0.94,0.86)	(0.93,0.82)
0.9	10	(0.93,0.76)	(0.93,0.73)	(0.92,0.69)	(0.89,0.64)
	20	(0.95,0.89)	(0.94,0.86)	(0.93,0.82)	(0.92,0.77)
	30	(0.95,0.92)	(0.94,0.89)	(0.94,0.85)	(0.92,0.81)
	40	(0.95,0.92)	(0.94,0.89)	(0.94,0.86)	(0.93,0.82)
	50	(0.95,0.93)	(0.95,0.90)	(0.94,0.86)	(0.93,0.82)

A.1.3 X-Uncertainty Convex Hull Method

Ω	0.1			
$channel\backslash\bar{\alpha}$	0.05	0.1	0.15	0.2
1	(0.97,0.96)	(0.97,0.94)	(0.97,0.92)	(0.97,0.89)
2	(0.95,0.94)	(0.95,0.92)	(0.95,0.91)	(0.95,0.88)
3	(0.98,0.96)	(0.98,0.93)	(0.98,0.90)	(0.98,0.85)
4	(0.91,0.91)	(0.91,0.90)	(0.91,0.88)	(0.91,0.87)

Ω	0.3			
$channel\backslash\bar{\alpha}$	0.05	0.1	0.15	0.2
1	(0.97,0.95)	(0.97,0.93)	(0.97,0.91)	(0.97,0.87)
2	(0.95,0.94)	(0.95,0.92)	(0.95,0.89)	(0.95,0.87)
3	(0.98,0.96)	(0.98,0.92)	(0.98,0.88)	(0.98,0.82)
4	(0.91,0.90)	(0.91,0.89)	(0.91,0.88)	(0.91,0.86)

Ω	0.5			
$channel\backslash\bar{\alpha}$	0.05	0.1	0.15	0.2
1	(0.97,0.95)	(0.97,0.92)	(0.97,0.88)	(0.97,0.83)
2	(0.95,0.93)	(0.95,0.90)	(0.95,0.87)	(0.95,0.82)
3	(0.98,0.95)	(0.98,0.91)	(0.98,0.85)	(0.98,0.77)
4	(0.91,0.90)	(0.91,0.88)	(0.91,0.85)	(0.91,0.82)
Ω	0.7			
$channel\backslash\bar{\alpha}$	0.05	0.1	0.15	0.2
1	(0.97,0.95)	(0.97,0.91)	(0.97,0.87)	(0.97,0.82)
2	(0.95,0.93)	(0.95,0.90)	(0.95,0.86)	(0.95,0.81)
3	(0.98,0.95)	(0.98,0.90)	(0.98,0.84)	(0.98,0.77)
4	(0.91,0.90)	(0.91,0.87)	(0.91,0.84)	(0.91,0.81)
Ω	0.9			
$channel\backslash\bar{\alpha}$	0.05	0.1	0.15	0.2
1	(0.97,0.94)	(0.97,0.91)	(0.97,0.87)	(0.97,0.82)
2	(0.95,0.92)	(0.95,0.89)	(0.95,0.85)	(0.95,0.81)
3	(0.98,0.95)	(0.98,0.90)	(0.98,0.84)	(0.98,0.77)
4	(0.91,0.89)	(0.91,0.87)	(0.91,0.84)	(0.91,0.80)

A.1.4 X- and Y-Uncertainty Nominal R-Squared Values

Ω	$gr\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
0.1	10	(0.93,0.56)	(0.93,0.56)	(0.93,0.56)	(0.93,0.56)	(0.93,0.56)
	20	(0.95,0.58)	(0.95,0.62)	(0.95,0.60)	(0.95,0.59)	(0.95,0.57)
	30	(0.95,0.61)	(0.95,0.66)	(0.95,0.65)	(0.95,0.63)	(0.95,0.61)
	40	(0.95,0.63)	(0.95,0.70)	(0.95,0.69)	(0.95,0.67)	(0.95,0.64)
	50	(0.95,0.79)	(0.95,0.83)	(0.95,0.82)	(0.95,0.80)	(0.95,0.75)
0.3	10	(0.93,0.56)	(0.93,0.56)	(0.93,0.56)	(0.93,0.56)	(0.93,0.56)
	20	(0.95,0.58)	(0.95,0.62)	(0.95,0.62)	(0.95,0.60)	(0.95,0.59)
	30	(0.95,0.60)	(0.95,0.66)	(0.95,0.66)	(0.95,0.64)	(0.95,0.62)
	40	(0.95,0.63)	(0.95,0.70)	(0.95,0.71)	(0.95,0.68)	(0.95,0.66)
	50	(0.95,0.79)	(0.95,0.83)	(0.95,0.83)	(0.95,0.81)	(0.95,0.76)
0.5	10	(0.93,0.54)	(0.93,0.54)	(0.93,0.54)	(0.93,0.54)	(0.93,0.54)
	20	(0.95,0.57)	(0.95,0.62)	(0.95,0.63)	(0.95,0.61)	(0.95,0.59)
	30	(0.95,0.59)	(0.95,0.66)	(0.95,0.68)	(0.95,0.66)	(0.95,0.63)
	40	(0.95,0.61)	(0.95,0.70)	(0.95,0.72)	(0.95,0.70)	(0.95,0.67)
	50	(0.95,0.77)	(0.95,0.83)	(0.95,0.83)	(0.95,0.82)	(0.95,0.79)
0.7	10	(0.93,0.53)	(0.93,0.53)	(0.93,0.52)	(0.93,0.52)	(0.93,0.52)
	20	(0.95,0.55)	(0.95,0.62)	(0.95,0.66)	(0.95,0.65)	(0.95,0.62)
	30	(0.95,0.57)	(0.95,0.66)	(0.95,0.69)	(0.95,0.69)	(0.95,0.67)
	40	(0.95,0.60)	(0.95,0.70)	(0.95,0.73)	(0.95,0.73)	(0.95,0.72)
	50	(0.95,0.75)	(0.95,0.82)	(0.95,0.82)	(0.95,0.82)	(0.95,0.81)
0.9	10	(0.93,0.44)	(0.93,0.44)	(0.93,0.44)	(0.93,0.43)	(0.93,0.44)
	20	(0.95,0.46)	(0.95,0.56)	(0.95,0.61)	(0.95,0.63)	(0.95,0.62)
	30	(0.95,0.48)	(0.95,0.59)	(0.95,0.64)	(0.95,0.66)	(0.95,0.66)
	40	(0.95,0.50)	(0.95,0.62)	(0.95,0.67)	(0.95,0.69)	(0.95,0.70)
	50	(0.95,0.64)	(0.95,0.72)	(0.95,0.74)	(0.95,0.76)	(0.95,0.77)

A.1.5 X- and Y-Uncertainty Robust R-Squared Values

Ω	$gr\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
0.1	10	(0.91,0.63)	(0.90,0.63)	(0.89,0.63)	(0.89,0.63)	(0.88,0.63)
	20	(0.91,0.68)	(0.90,0.71)	(0.89,0.70)	(0.89,0.68)	(0.88,0.67)
	30	(0.91,0.70)	(0.90,0.77)	(0.89,0.78)	(0.89,0.73)	(0.89,0.70)
	40	(0.91,0.72)	(0.91,0.83)	(0.90,0.86)	(0.90,0.77)	(0.90,0.73)
	50	(0.91,0.84)	(0.91,0.90)	(0.91,0.91)	(0.91,0.91)	(0.91,0.91)
0.3	10	(0.92,0.63)	(0.90,0.63)	(0.88,0.63)	(0.87,0.63)	(0.86,0.63)
	20	(0.93,0.69)	(0.90,0.72)	(0.88,0.71)	(0.87,0.70)	(0.88,0.68)
	30	(0.93,0.70)	(0.91,0.78)	(0.90,0.79)	(0.90,0.75)	(0.91,0.72)
	40	(0.93,0.72)	(0.92,0.84)	(0.92,0.87)	(0.93,0.81)	(0.93,0.75)
	50	(0.93,0.83)	(0.93,0.89)	(0.93,0.92)	(0.93,0.91)	(0.93,0.91)
0.5	10	(0.92,0.62)	(0.90,0.62)	(0.87,0.62)	(0.86,0.62)	(0.85,0.62)
	20	(0.92,0.68)	(0.88,0.72)	(0.87,0.73)	(0.87,0.71)	(0.92,0.68)
	30	(0.92,0.69)	(0.90,0.78)	(0.90,0.80)	(0.90,0.78)	(0.92,0.72)
	40	(0.92,0.71)	(0.92,0.84)	(0.92,0.87)	(0.93,0.85)	(0.93,0.76)
	50	(0.93,0.82)	(0.93,0.89)	(0.93,0.92)	(0.93,0.91)	(0.93,0.91)
0.7	10	(0.92,0.59)	(0.90,0.59)	(0.87,0.59)	(0.85,0.59)	(0.84,0.59)
	20	(0.93,0.65)	(0.88,0.71)	(0.86,0.73)	(0.91,0.70)	(0.92,0.69)
	30	(0.93,0.67)	(0.90,0.76)	(0.90,0.79)	(0.92,0.78)	(0.93,0.75)
	40	(0.93,0.69)	(0.93,0.80)	(0.93,0.85)	(0.93,0.85)	(0.93,0.82)
	50	(0.93,0.80)	(0.93,0.87)	(0.93,0.90)	(0.93,0.91)	(0.93,0.91)
0.9	10	(0.92,0.50)	(0.89,0.50)	(0.86,0.50)	(0.85,0.50)	(0.84,0.50)
	20	(0.93,0.55)	(0.88,0.62)	(0.87,0.65)	(0.92,0.66)	(0.93,0.65)
	30	(0.93,0.56)	(0.90,0.65)	(0.90,0.69)	(0.92,0.70)	(0.93,0.70)
	40	(0.93,0.58)	(0.93,0.67)	(0.93,0.72)	(0.93,0.75)	(0.93,0.75)
	50	(0.93,0.68)	(0.93,0.74)	(0.93,0.79)	(0.93,0.81)	(0.93,0.82)

A.2 Real-Life Data

A.2.1 X-Uncertainty Nominal R-Squared Values

$channel \backslash \Omega$	0.1	0.3	0.5	0.7	0.9
1	(0.12,0.04)	(0.12,0.01)	(0.12,0.02)	(0.12,0.02)	(0.12,0.02)
2	(0.17,0.14)	(0.17,0.10)	(0.17,0.10)	(0.17,0.10)	(0.17,0.09)
3	(0.10,0.07)	(0.10,0.05)	(0.10,0.05)	(0.10,0.05)	(0.10,0.05)
4	(0.12,0.08)	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)
5	(0.14,0.05)	(0.14,0.05)	(0.14,0.04)	(0.14,0.03)	0.14,0.03)
6	(0.10,0.07)	(0.10,0.05)	(0.10,0.05)	(0.10,0.05)	(0.10,0.05)
7	(0.40,0.33)	(0.40,0.24)	(0.40,0.23)	(0.40,0.22)	(0.40,0.21)
8	(0.21,0.16)	(0.21,0.10)	(0.21,0.09)	(0.21,0.08)	(0.21,0.08)
9	(0.22,0.16)	(0.22,0.13)	(0.22,0.13)	(0.22,0.12)	(0.22,0.12)
10	(0.10,0.05)	(0.10,0.04)	(0.10,0.04)	(0.10,0.04)	(0.10,0.04)

A.2.2 X-Uncertainty Robust R-Squared Values

$channel \backslash \Omega$	0.1	0.3	0.5	0.7	0.9
1	(0.10,0.06)	(0.10,0.04)	(0.10,0.04)	(0.10,0.04)	(0.10,0.04)
2	(0.16,0.14)	(0.16,0.12)	(0.16,0.11)	(0.16,0.11)	(0.16,0.11)
3	(0.09,0.07)	(0.09,0.06)	(0.09,0.06)	(0.09,0.06)	(0.09,0.06)
4	(0.11,0.09)	(0.11,0.08)	(0.10,0.08)	(0.10,0.08)	(0.10,0.08)
5	(0.12,0.09)	(0.12,0.09)	(0.12,0.09)	(0.12,0.09)	(0.12,0.09)
6	(0.08,0.07)	(0.08,0.06)	(0.08,0.06)	(0.09,0.06)	(0.09,0.06)
7	(0.36,0.31)	(0.35,0.26)	(0.36,0.26)	(0.36,0.25)	(0.36,0.25)
8	(0.17,0.15)	(0.16,0.13)	(0.16,0.13)	(0.16,0.13)	(0.17,0.13)
9	(0.21,0.16)	(0.20,0.14)	(0.20,0.14)	(0.20,0.14)	(0.20,0.14)
10	(0.08,0.07)	(0.08,0.06)	(0.08,0.06)	(0.08,0.06)	(0.08,0.06)

A.2.3 X-Uncertainty Convex Hull Method

$channel \backslash \Omega$	0.1	0.3	0.5	0.7	0.9
1	(-0.03,-0.04)	(-0.03,-0.04)	(-0.03,-0.05)	(-0.03,-0.05)	(-0.03,-0.05)
2	(0.14,0.10)	(0.14,0.09)	(0.14,0.08)	(0.14,0.07)	(0.14,0.07)
3	(-0.08,-0.08)	(-0.08,-0.09)	(-0.08,-0.09)	(-0.08,-0.09)	(-0.08,-0.09)
4	(0.00,0.00)	(0.00,-0.01)	(0.00,-0.01)	(0.00,-0.01)	(0.00,-0.01)
5	(0.11,0.10)	(0.11,0.10)	(0.11,0.10)	(0.11,0.09)	(0.11,0.09)
6	(0.08,0.06)	(0.08,0.05)	(0.08,0.04)	(0.08,0.04)	(0.08,0.04)
7	(0.39,0.29)	(0.39,0.27)	(0.39,0.22)	(0.39,0.20)	(0.39,0.18)
8	(0.20,0.15)	(0.20,0.13)	(0.20,0.10)	(0.20,0.09)	(0.20,0.08)
9	(0.13,0.09)	(0.13,0.08)	(0.13,0.07)	(0.13,0.07)	(0.13,0.07)
10	(-0.08,-0.08)	(-0.08,-0.08)	(-0.08,-0.09)	(-0.08,-0.09)	(-0.08,-0.09)

A.2.4 X- and Y-Uncertainty Nominal R-Squared Values

Ω	0.1				
$channel\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.12,0.01)	(0.12,0.06)	(0.12,0.07)	(0.12,0.07)	(0.12,0.07)
2	(0.17,0.10)	(0.17,0.16)	(0.17,0.16)	(0.17,0.16)	(0.17,0.15)
3	(0.10,0.05)	(0.10,0.10)	(0.10,0.10)	(0.10,0.10)	(0.10,0.10)
4	(0.12,0.06)	(0.12,0.08)	(0.12,0.09)	(0.12,0.11)	(0.12,0.11)
5	(0.14,-0.01)	(0.14,0.00)	(0.14,0.00)	(0.14,-0.01)	(0.14,-0.01)
6	(0.10,0.07)	(0.10,0.08)	(0.10,0.08)	(0.10,0.08)	(0.10,0.07)
7	(0.40,0.14)	(0.40,0.28)	(0.40,0.27)	(0.40,0.22)	(0.40,0.17)
8	(0.21,0.08)	(0.21,0.14)	(0.21,0.14)	(0.21,0.08)	(0.21,0.02)
9	(0.22,0.13)	(0.22,0.21)	(0.22,0.21)	(0.22,0.21)	(0.22,0.21)
10	(0.10,0.05)	(0.10,0.10)	(0.10,0.10)	(0.10,0.10)	(0.10,0.10)
Ω	0.3				
$channel\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.12,0.00)	(0.12,0.03)	(0.12,0.03)	(0.12,0.05)	(0.12,0.05)
2	(0.17,0.08)	(0.17,0.11)	(0.17,0.11)	(0.17,0.11)	(0.17,0.10)
3	(0.10,0.06)	(0.10,0.08)	(0.10,0.09)	(0.10,0.09)	(0.10,0.09)
4	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)	(0.12,0.07)	(0.12,0.07)
5	(0.14,0.00)	(0.14,0.00)	(0.14,0.00)	(0.14,-0.01)	(0.14,0.00)
6	(0.10,0.03)	(0.10,0.05)	(0.10,0.05)	(0.10,0.03)	(0.10,0.04)
7	(0.40,0.05)	(0.40,0.09)	(0.40,0.06)	(0.40,0.00)	(0.40,-0.02)
8	(0.21,0.03)	(0.21,0.00)	(0.21,-0.02)	(0.21,0.00)	(0.21,0.00)
9	(0.22,0.08)	(0.22,0.19)	(0.22,0.20)	(0.22,0.20)	(0.22,0.20)
10	(0.10,0.05)	(0.10,0.07)	(0.10,0.09)	(0.10,0.10)	(0.10,0.10)
Ω	0.5				
$channel\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.12,0.00)	(0.12,0.01)	(0.12,0.01)	(0.12,0.03)	(0.12,0.03)
2	(0.17,0.06)	(0.17,0.10)	(0.17,0.10)	(0.17,0.10)	(0.17,0.10)
3	(0.10,0.06)	(0.10,0.07)	(0.10,0.08)	(0.10,0.08)	(0.10,0.08)
4	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)
5	(0.14,0.00)	(0.14,-0.01)	(0.14,-0.01)	(0.14,-0.01)	(0.14,-0.01)
6	(0.10,0.03)	(0.10,0.04)	(0.10,0.04)	(0.10,0.03)	(0.10,0.03)
7	(0.40,0.03)	(0.40,0.02)	(0.40,0.00)	(0.40,-0.03)	(0.40,-0.03)
8	(0.21,0.00)	(0.21,-0.03)	(0.21,-0.03)	(0.21,-0.03)	(0.21,0.00)
9	(0.22,0.07)	(0.22,0.16)	(0.22,0.17)	(0.22,0.17)	(0.22,0.17)
10	(0.10,0.06)	(0.10,0.07)	(0.10,0.07)	(0.10,0.08)	(0.10,0.08)

Ω	0.7				
$channel\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.12,0.00)	(0.12,0.00)	(0.12,0.01)	(0.12,0.01)	(0.12,0.01)
2	(0.17,0.05)	(0.17,0.09)	(0.17,0.10)	(0.17,0.10)	(0.17,0.10)
3	(0.10,0.06)	(0.10,0.06)	(0.10,0.07)	(0.10,0.07)	(0.10,0.06)
4	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)	(0.12,0.06)
5	(0.14,-0.01)	(0.14,-0.01)	(0.14,-0.01)	(0.14,-0.02)	(0.14,-0.02)
6	(0.10,0.03)	(0.10,0.03)	(0.10,0.03)	(0.10,0.03)	(0.10,0.03)
7	(0.40,0.02)	(0.40,-0.04)	(0.40,-0.05)	(0.40,-0.05)	(0.40,-0.04)
8	(0.21,0.02)	(0.21,-0.05)	(0.21,-0.03)	(0.21,-0.02)	(0.21,0.00)
9	(0.22,0.06)	(0.22,0.14)	(0.22,0.15)	(0.22,0.14)	(0.22,0.13)
10	(0.10,0.05)	(0.10,0.06)	(0.10,0.06)	(0.10,0.07)	(0.10,0.06)
Ω	0.9				
$channel\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.12,0.00)	(0.12,0.00)	(0.12,0.00)	(0.12,0.00)	(0.12,0.00)
2	(0.17,0.05)	(0.17,0.05)	(0.17,0.05)	(0.17,0.05)	(0.17,0.05)
3	(0.10,0.04)	(0.10,0.04)	(0.10,0.04)	(0.10,0.04)	(0.10,0.04)
4	(0.12,0.05)	(0.12,0.05)	(0.12,0.05)	(0.12,0.05)	(0.12,0.05)
5	(0.14,-0.08)	(0.14,-0.08)	(0.14,-0.08)	(0.14,-0.08)	(0.14,-0.08)
6	(0.10,0.01)	(0.10,0.01)	(0.10,0.01)	(0.10,0.01)	(0.10,0.01)
7	(0.40,0.03)	(0.40,0.03)	(0.40,0.03)	(0.40,-0.05)	(0.40,-0.05)
8	(0.21,0.00)	(0.21,0.00)	(0.21,0.00)	(0.21,-0.03)	(0.21,-0.03)
9	(0.22,0.06)	(0.22,0.06)	(0.22,0.06)	(0.22,0.06)	(0.22,0.06)
10	(0.10,0.02)	(0.10,0.02)	(0.10,0.02)	(0.10,0.02)	(0.10,0.02)

A.2.5 X- and Y-Uncertainty Robust R-Squared Values

Ω	0.1				
$channel\backslash\gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.11,0.01)	(0.11,0.07)	(0.10,0.09)	(0.10,0.10)	(0.10,0.09)
2	(0.16,0.10)	(0.16,0.15)	(0.18,0.16)	(0.19,0.16)	(0.18,0.15)
3	(0.09,0.09)	(0.11,0.09)	(0.12,0.08)	(0.12,0.09)	(0.12,0.09)
4	(0.11,0.07)	(0.10,0.09)	(0.11,0.10)	(0.12,0.11)	(0.12,0.11)
5	(0.12,0.12)	(0.12,0.12)	(0.13,0.12)	(0.13,0.11)	(0.14,0.11)
6	(0.08,0.07)	(0.11,0.07)	(0.12,0.07)	(0.13,0.06)	(0.13,0.07)
7	(0.34,0.28)	(0.39,0.27)	(0.42,0.24)	(0.42,0.20)	(0.41,0.18)
8	(0.16,0.15)	(0.23,0.12)	(0.24,0.10)	(0.25,0.07)	(0.24,0.06)
9	(0.20,0.15)	(0.23,0.19)	(0.25,0.18)	(0.25,0.19)	(0.25,0.18)
10	(0.09,0.06)	(0.11,0.08)	(0.12,0.08)	(0.12,0.08)	(0.12,0.09)

Ω	0.3				
$channel \backslash \gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.10,0.01)	(0.10,0.04)	(0.11,0.04)	(0.11,0.06)	(0.11,0.06)
2	(0.15,0.10)	(0.15,0.11)	(0.15,0.12)	(0.15,0.13)	(0.15,0.12)
3	(0.09,0.08)	(0.10,0.09)	(0.10,0.09)	(0.10,0.09)	(0.10,0.09)
4	(0.11,0.07)	(0.11,0.07)	(0.11,0.09)	(0.11,0.09)	(0.11,0.09)
5	(0.12,0.12)	(0.13,0.12)	(0.14,0.11)	(0.14,0.11)	(0.15,0.11)
6	(0.08,0.07)	(0.11,0.06)	(0.10,0.06)	(0.10,0.06)	(0.08,0.07)
7	(0.31,0.17)	(0.39,0.17)	(0.35,0.16)	(0.33,0.15)	(0.35,0.15)
8	(0.15,0.09)	(0.23,0.05)	(0.19,0.06)	(0.14,0.01)	(0.15,0.04)
9	(0.20,0.11)	(0.22,0.19)	(0.22,0.18)	(0.21,0.18)	(0.22,0.18)
10	(0.09,0.05)	(0.10,0.09)	(0.10,0.09)	(0.10,0.09)	(0.11,0.09)
Ω	0.5				
$channel \backslash \gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.10,0.01)	(0.10,0.03)	(0.10,0.03)	(0.11,0.04)	(0.10,0.04)
2	(0.15,0.10)	(0.15,0.11)	(0.15,0.11)	(0.15,0.12)	(0.15,0.11)
3	(0.09,0.08)	(0.10,0.09)	(0.10,0.09)	(0.10,0.08)	(0.10,0.09)
4	(0.11,0.07)	(0.11,0.07)	(0.11,0.08)	(0.11,0.08)	(0.11,0.08)
5	(0.12,0.12)	(0.13,0.12)	(0.13,0.12)	(0.13,0.12)	(0.13,0.12)
6	(0.08,0.05)	(0.09,0.07)	(0.09,0.05)	(0.09,0.05)	(0.08,0.05)
7	(0.31,0.15)	(0.36,0.15)	(0.34,0.14)	(0.33,0.14)	(0.34,0.11)
8	(0.14,0.06)	(0.18,0.03)	(0.17,0.06)	(0.14,0.04)	(0.15,0.05)
9	(0.20,0.09)	(0.21,0.17)	(0.21,0.17)	(0.21,0.15)	(0.21,0.15)
10	(0.09,0.05)	(0.09,0.07)	(0.10,0.08)	(0.10,0.08)	(0.10,0.08)
Ω	0.7				
$channel \backslash \gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.10,0.01)	(0.10,0.01)	(0.10,0.02)	(0.10,0.02)	(0.10,0.02)
2	(0.15,0.10)	(0.15,0.10)	(0.15,0.11)	(0.15,0.11)	(0.15,0.10)
3	(0.09,0.08)	(0.09,0.08)	(0.09,0.08)	(0.09,0.08)	(0.09,0.08)
4	(0.11,0.07)	(0.11,0.07)	(0.11,0.07)	(0.11,0.07)	(0.11,0.07)
5	(0.12,0.12)	(0.12,0.12)	(0.13,0.12)	(0.13,0.12)	(0.13,0.12)
6	(0.08,0.04)	(0.08,0.07)	(0.08,0.04)	(0.08,0.04)	(0.08,0.04)
7	(0.31,0.12)	(0.32,0.13)	(0.33,0.12)	(0.33,0.12)	(0.33,0.10)
8	(0.14,0.03)	(0.13,0.01)	(0.14,0.06)	(0.15,0.07)	(0.14,0.06)
9	(0.20,0.08)	(0.20,0.15)	(0.20,0.16)	(0.21,0.13)	(0.21,0.12)
10	(0.09,0.05)	(0.09,0.06)	(0.09,0.07)	(0.09,0.07)	(0.09,0.06)

Ω	0.9				
$channel \backslash \gamma$	0.1	0.3	0.5	0.7	0.9
1	(0.10,0.01)	(0.10,0.01)	(0.10,0.01)	(0.10,0.01)	(0.10,0.01)
2	(0.15,0.10)	(0.15,0.07)	(0.15,0.07)	(0.15,0.07)	(0.15,0.07)
3	(0.09,0.08)	(0.09,0.05)	(0.09,0.05)	(0.09,0.05)	(0.09,0.05)
4	(0.11,0.07)	(0.11,0.07)	(0.11,0.07)	(0.11,0.07)	(0.11,0.07)
5	(0.12,0.12)	(0.12,0.08)	(0.12,0.08)	(0.12,0.08)	(0.12,0.08)
6	(0.08,0.03)	(0.08,0.03)	(0.08,0.03)	(0.08,0.03)	(0.08,0.03)
7	(0.31,0.12)	(0.33,0.09)	(0.33,0.09)	(0.33,0.09)	(0.33,0.09)
8	(0.14,0.03)	(0.14,0.06)	(0.14,0.06)	(0.14,0.06)	(0.14,0.06)
9	(0.20,0.08)	(0.20,0.05)	(0.20,0.05)	(0.20,0.05)	(0.20,0.05)
10	(0.09,0.05)	(0.09,0.02)	(0.09,0.02)	(0.09,0.02)	(0.09,0.02)

A.3 End-Model Results

$Reg. \backslash B$	Ω	γ	50	100	150	200	250	300	350	400
Nominal			496.56	717.8	868.02	992.29	1116.56	1240.82	1340.05	1347.26
Rob.X	0.3		559.74	751.39	837.19	898.59	951.79	1004.67	1039.11	1040.38
	0.9		555.14	753.62	840.66	900.97	950.94	1000.91	1035.38	1035.38
Rob.XY	0.3	0.1	609.34	813.06	892.48	936.07	975.62	1015.17	1049.52	1050.88
		0.5	541.59	840.62	1027.63	1180.00	1285.80	1300.67	1300.67	1300.67
		0.9	587.37	802.97	892.07	957.31	1008.38	1058.63	1094.83	1094.86
	0.9	0.1	591.49	786.15	870.97	911.43	950.63	988.38	1005.02	1005.02
		0.5	567.49	777.15	880.97	942.43	987.63	1042.09	1089.12	1113.37
		0.9	602.79	739.15	849.34	902.43	938.26	964.49	1003.67	1003.69
CHM			643.70	864.10	962.77	1033.37	1094.56	1155.37	1194.97	1196.44

B Convexity and Concavity

A real-valued function f on an interval I is said to be concave if

$$f((1-t)x+ty) \geq (1-t)f(x)+tf(y) \;\; \forall x,y \in I, \; t \in [0,1]$$

holds true. The definition of a convex function is similar, but with the inequality reversed, i.e.

$$f((1-t)x+(t)y) \leq (1-t)f(x)+(t)f(y) \;\; \forall x,y \in I, \; t \in [0,1]$$

it follows directly that a function f is convex on I if and only if the function -f is concave on I. One can deduce the theorem that a sum over convex functions is again convex as follows:

Assume f and g are two convex functions on I, $x,y \in I$ *and* $t \in [0,1]$

$$(f+g)(tx+(1-t)y) = f(tx+(1-t)y) + g(tx+(1-t)y)$$

$$\leq tf(x)+(1-t)f(y)+tg(x)+(1-t)g(y) = t(f+g)(x)+(1-t)(f+g)(y)$$