# Methods for Robustness Programming

Niels J. Olieman

Promotor
prof.dr. P. van Beek
Emeritus Hoogleraar Operationele Research en Logistiek,
Wageningen Universiteit

Co-promotor
dr. E.M.T. Hendrix
Universitair Hoofddocent, leerstoelgroep Operationele Research
en Logistiek, Wageningen Universiteit

Promotiecommissie
prof.ir. A.J.M. Beulens, Wageningen Universiteit, Nederland
prof.dr. I. Deák, Corvinus University of Budapest, Hungary
dr.ir. G.M.P. van Kempen, Unilever R&D, Vlaardingen, Nederland
prof.dr. C. Roos, Technische Universiteit Delft, Nederland

*Dit onderzoek is uitgevoerd binnen de Mansholt Graduate School of Social Sciences*

# Methods for Robustness Programming

Nicolaas Johannes Olieman

Proefschrift
ter verkrijging van de graad van doctor
op gezag van de rector magnificus
van Wageningen Universiteit,
prof.dr. M.J. Kropff,
in het openbaar te verdedigen
op woensdag 12 maart 2008
des namiddags te half twee in de Aula.

Methods for Robustness Programming
PhD thesis Wageningen University - with references - with summaries in English and Dutch
Olieman, N.J., 2008
ISBN 978-90-8504-876-3

# Preface

To me, mathematics is associated with a colourful abstract world in which all valid theorems exist and of which only an infinitely small fraction has been discovered yet. The discovery of parts of this abstract world that lead to methods for solving practical problems, plays a central role in my ambitions. In this process of doing research, I am intrigued by a principle nicely put by the Japanese poet Matsuo Basho (1644-1694): 'Do not follow in the footsteps of the old masters, but seek what they sought'.

I started this journey more than 5 years ago and that would not have been possible without the support and guidance of Eligius Hendrix and Paul van Beek and I like to thank them for sharing their insights with me, the many joyful discussions and their efforts that contributed to the published articles and this thesis.

I am grateful to the Operations Research and Logistics (ORL) group for facilitating this research. I like to thank the ORL colleagues Jack van der Vorst, Frits Claassen, Theo Hendriks, Joke van Lemmen-Gerdessen, Karin Pauls-Worm, Elena Sáiz-Pérez and last but not least Ria Slootman-Vermeer for the very good time I have had while working in Wageningen and the many times I could share my joys or find support in less happy circumstances. I also like to thank the colleagues of the Information Technology Group who often joined us and were part of the inspiring coffee break discussions.

I thank Ekko van Ierland of the Environmental Economics and Natural Resources (ENR) group for the support of my research and allowing me to be part of the STACO research project. In this respect I am thankful to Rob Dellink, Michael Finus, Hans-Peter Weikard and Juan-Carlos Altamirano-Cabrera, for their contributions to the articles we wrote and the many things I have learned from our cooperation at the early stages of my research.

I am thankful to Bram van Putten for sharing his mathematical passion for Probability Theory and Measure Theory and for convincing me in 1992 to start my studies in Wageningen.

An important part of this thesis is based on case studies of mixture designs for Unilever R&D. I like to thank Geert van Kempen for facilitating the student-internships in Vlaardingen, sharing and explaining the case studies with us and particularly about the fact that -despite his demanding work- he always found time to continue the cooperation. I also like to thank the students, Annet Bastiaanse and Saskia van der Leer, for writing two nice MSc. theses, which were very useful for my research.

I am aware that in the last year, my full-time job at Rabobank Netherlands in combination with finalising this thesis, reduced the time I could spend with my friends. I like to thank my friends at the Stichting Aikido Kyorakukan, Het Bourgondisch Dispuut and all those I forgot, for their patience and understanding.

I want to thank my family for the various ways they contributed to this thesis. I am grateful to my parents for their unconditional support. I want to thank Sebastiaan for accepting me as a semi-dad and being considerate during busy times. Finally, I want to thank Majda for her understanding and support when I had to work through the weekends, for always being there for me and just because she lets me love her.

Niels Olieman, March 2008

# Contents

# Chapter 1
# Introduction

## 1.1 Robustness

Consider a wooden table with 4 table legs, each with a diameter of 20 cm and a table top that is 10 cm thick. There are 10 persons sitting at the table, telling jokes and having a good time with beer and wine. Suppose that the *uncontrollable factors* such as bumping and pushing against the table, hardly moves the table. The table can be called robust, since the desirable table properties such as *stability* and *maximum weight support* appear not sensitive to the considered uncertainties. Consider a similar table, where the mentioned dimensions are twice as big. The latter table is probably *more* robust. Apparently an object can have some level of *Robustness*, since some objects can be more robust than others.

Although the described wooden table is robust in the context of a party, a plastic table is probably more robust in the context of outdoors weather conditions. The Robustness of an object's required properties, is context and purpose dependent.

As another example, consider a container of yoghurt. Typically the container is stored in the refrigerator, but is warming up towards room temperature when it is outside the refrigerator. The daily cycle of cooling down and warming up can have an effect on the structure of the yoghurt, such that over time the yoghurt does not look appealing anymore. A yoghurt product which appears not to be affected by temperature fluctuations can be called robust. A yoghurt product which maintains the desirable properties for the widest range of temperature fluctuations has the highest Robustness.

The concept of Robustness is often associated to physical objects. In principle the Robustness concept can also be associated to non-physical objects. Consider a government policy as the subject of interest. A government can *design* a policy for reducing $CO_2$ emissions, such that the emissions in 2008 until 2012 are on average 6% lower than in 1990. In this policy-making it is sensible to take into account all sorts of uncertainties, for instance related to climate change and economic growth. Under the assumption that there are alternative policies to reach the goal of 6% emission reduction, it is interesting to know which policy is least sensitive to the perceived uncertain factors, i.e. which policy is most robust.

Another example of a non-physical object is a long-term transportation plan. In general, the motivation to design a transportation plan is to satisfy supply and demand quantities in time at minimal cost. While designing a transportation plan for the long-term, it is sensible to take into account uncertainties in traffic congestion, fuel prices and customer demand. In this context, the Robustness of a transportation plan is two fold: (i) the Robustness of satisfying the demand, supply and timing *constraints* ; (ii) the Robustness of minimal total cost *objective.*

Consider object qualities such as *safety, reliability, stability* and *security.* All these object qualities can be seen as to what extent an object maintains required properties under influence of uncertain factors. For the general case, Robustness of an object is a measure for the insensitivity of required object properties to uncertain factors. In Chapter 2, the Framework of Robustness Programming is defined. The framework is intended as a generalisation for the mathematical definition of Robustness optimisation problems, inspired by similar problem formulations as can be found in Nie and Ellingwood (2000); Franchin et al. (2002); Parkinson et al. (1993); Du and Chen (2000).

## 1.2 Probabilistic Robustness

Robustness is related to uncertain uncontrollable factors. There exist alternative modelling paradigms for modelling uncertainty and each gives rise to alternative measures for Robustness. For instance, Taguchi (1986) and Markowitz (1952) implicitly model Robustness by only considering the *mean* and *variance* statistics. Ben-Tal and Nemirovski (2002) use a binary approach: an object is either robust or not, i.e. an object either maintains its properties as required under all possible conditions or not.

A common approach in science for modelling uncertainty, is studied in the science field called *Probability Theory* (Ghahramani, 2000; Grimmett and Stirzaker, 2001; Jacod and Protter, 2004). In Probability Theory, the notions *random vector* and *probability space* are formally introduced, which can be used to model a collection of uncertain events. The general notation conventions used in this thesis can be found in Appendix A.1 and a formal embedding in Probability Theory of the concepts such as the sample space $\mathbb{V}$, the $\sigma$-field $\mathcal{V}$ and probability measure $\Pr_{\boldsymbol{v}}$ related to random vector $\boldsymbol{v}$, can be found in Appendix A.2.

In this theory $\boldsymbol{v}$ is a random vector defined on probability space $(\mathbb{V}, \mathcal{V}, \Pr_{\boldsymbol{v}})$ with possible realisations $v \in \mathbb{V} \subseteq \mathbb{R}^{\mathrm{N}}$. The typical use is to assign a probability $\Pr_{\boldsymbol{v}}(\mathbb{S})$ to the event that the random vector $\boldsymbol{v}$ will have realisations in some set $\mathbb{S} \in \mathcal{V}$, $\mathbb{S} \subseteq \mathbb{R}^{\mathrm{N}}$. Two equivalent notations are used $\Pr_{\boldsymbol{v}}(\mathbb{S}) = \Pr\{\boldsymbol{v} \in \mathbb{S}\}$.

The use of Probability Theory in the context of Robustness is the following. Consider that uncontrollable factors, which affect object properties, are perceived as a random vector $\boldsymbol{v}$. A modelling approach is introduced in Chapter 2, which makes it possible to verify for each realisation $v \in \mathbb{V}$, whether or not the realisation results in desirable object properties. The realisations $v$ for which the object has properties as required, can be collected in a set $\mathbb{H} \subseteq \mathbb{V}$. Now it is possible to express Robustness in a probabilistic way: $R = \Pr\{\boldsymbol{v} \in \mathbb{H}\}$. In this thesis, probabilistic Robustness is studied and is defined as:

**Definition 1.1** *Robustness of an object is the probability that an object will have properties as required.*

## 1.3 Designing and Robustness optimisation

In design processes in industries, the intention is to create consumer products with properties that fulfil product specifications. If product properties are uncertain, then it is also uncertain whether these properties fulfil the specifications. This thesis studies methods that can be used to find product designs with a Robustness as big as possible such that products with uncertain properties fulfil specifications.

An example of an uncertain object property is the Number of Bacteria at the Best Before date (NBatBB) in a food product. The average temperature between the moment of production and moment of consumption has an influence on the growth rate of the bacteria population. Assume that only the growth rate and initial number of bacteria right after production determine the NBatBB property. It is technically and practically not tractable to count each individual bacterium. A practical alternative is to estimate the initial number of bacteria. Such estimate inevitability leads to some degree of uncertainty about the *true* initial number of bacteria. The average temperature of the food product after purchase and before consumption can also vary, depending on the distance to the grocery shop and car temperature. From a public health point of view, it is relevant to maximise the Robustness that the NBatBB will be below the health safety limit, by making good judgement about the involved uncertainties and choosing the Best Before (BB) date appropriately.

Two types of factors that influence object properties, can be distinguished: *controllable factors* and *uncontrollable factors*. The BB date is an example of a controllable factor, since the producer can choose the BB date that is printed on the food product. The initial number of bacteria and the average temperature are examples of uncontrollable factors. It is relevant to find the best controllable factor setting, resulting in a maximum or acceptable number of products with properties that fulfil specifications. Authors such as Du and Chen (2000), Bjerager (1988) and Nie and Ellingwood (2000), modelled uncontrollable factors as random vectors and obtained encouraging results with their Robustness optimisation approaches.

In the competition for producing most reliable, durable and safe products, the industrial sectors such as electronics, aviation, automotive, offshore, construction and food, show a great interest for producing robust products. In this context, mathematical methods for Robustness optimisation are useful.

The concept of Robustness optimisation can also be useful for other sectors: In the financial sector, attention is paid to optimising the Robustness of investment portfolios; in the public sector it is relevant to have optimal robust policies and in logistics it is relevant to optimise the Robustness of supply chains and transportation plans.

## 1.4 Robustness Programming

In this thesis, Robustness Programming is defined as a mathematical approach for defining Robustness, computing or estimating Robustness and to find optimal values for the controllable factors that optimise Robustness. Stochastic Programming (SP) has its roots in Mathematical Programming (MP) and following this line of evolution, the Robustness Programming (RP) concepts developed in this thesis are based on Stochastic Programming concepts. Robustness Programming mainly differs from Stochastic Programming in the solution concept. To illustrate this idea, consider the following MP, SP and RP approach for the maximisation of some profit function $f : \mathbb{R}^{\mathrm{I}} \times \mathbb{R}^{\mathrm{N}} \longrightarrow \mathbb{R}$. The objective function $f(x, v)$ depends on a controllable factor $x \in \mathbb{R}^{\mathrm{I}}$ and an uncontrollable factor $v \in \mathbb{R}^{\mathrm{N}}$. The uncertainty about the uncontrollable factor is modelled as a random vector $\boldsymbol{v}$. An approach is to assume constant values v=$E(\boldsymbol{v})$ and use MP to find

$f^* = \max_x f(x, \mathrm{v})$. A Stochastic Programming approach for this problem, is to maximise expected profit: $f^* = \max_x E_{\boldsymbol{v}} f(x, \boldsymbol{v})$. The Robustness Programming equivalent requires a conceptual extension, since the Robustness concept is relative to a target or goal. For instance, one can maximise the Robustness of not making a loss. In that case the goal is to have non-negative profit (i.e. a lower bound of zero). The maximum Robustness of not making a loss is $R^* = \max_x \Pr\{f(x, \boldsymbol{v}) \geq 0\}$. More general, one can maximise the Robustness of reaching any other profit goal $\gamma \in \mathbb{R}^{\mathrm{K}}$, K= 1 with Robustness maximum $R^{[\gamma]*} = \max_x \Pr\{f(x, \boldsymbol{v}) \geq \gamma\}$ and corresponding best decision(s) with $x^{[\gamma]*} = \arg\max_x \Pr\{f(x, \boldsymbol{v}) \geq \gamma\}$. The solution concept of Robustness Programming is not a single solution, but a set $\mathbb{P}$ called the Robustness Programming Set (RPS) with elements $(\gamma, x^{[\gamma]*}, R^{[\gamma]*}) \in \mathbb{P} \subseteq \mathbb{R}^{\mathrm{K+I+1}}$. In the profit maximisation case, each element of the set $\mathbb{P}$ shows the relation between profit goals $\gamma$, optimal robust decisions $x^{[\gamma]*}$ and corresponding optimal profit Robustness $R^{[\gamma]*}$.

In this section, the concepts of Robustness Programming are further illustrated via the so-called News Vendor Problem and a Mixture Design problem. The following is a preliminary introduction to Robustness Programming notation, which is elaborated in detail in Chapter 2: Controllable factors are modelled as a vector $x$ and uncontrollable factors are modelled as a random vector $\boldsymbol{v}$. The object properties are modelled as a function $u(x, v)$. The aim is to have object properties between bounds that depend on the so-called goal parameter $\gamma$, such that $L(\gamma) \leq u(x, v) \leq H(\gamma)$. The Robustness given parameter $\gamma$ is $R^{[\gamma]}(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}^{[\gamma]}(x)\right\}$, with $\mathbb{H}^{[\gamma]}(x) = \{v | L(\gamma) \leq u(x, v) \leq H(\gamma)\}$.

### 1.4.1 News Vendor Problem illustration

The classic News Vendor Problem as described by Silver et al. (1998) and Tijms (2002), deals with the decision on the optimal newspaper order quantity, given a customer demand probability distribution. In the science area known as Supply Chain Management, the News Vendor Problem is often presented as a metaphor for a whole class of optimal order quantity problems. The mathematical structure of the News Vendor Problem has general properties applicable in Supply Chain Management. The following illustrates the idea of optimal Expected value versus optimal Robustness.

In this example, the news vendor can buy newspapers from the publisher for €1.- each and sells these for €2.- each, thus making a profit of €1.- for each unit sold. The news vendor will not re-order when all newspapers are sold and unsold newspapers at the end of the day are worthless and do not imply deposition costs. Let $x$ be the number of newspapers the news vendor intends to order from the supplier and assume that the uncertain customer demand $\boldsymbol{v}$ has a probability distribution, such that all possible demands from $\mathbb{V} = \{100, 101, .., 199, 200\}$ have the same probability $\Pr(v) = \frac{1}{101}$ for all $v \in \mathbb{V}$. If the demand $v$ is less than or equal to the newspaper stock $x$, the news vendor revenues are €$2v - x$. If the demand $v$ is bigger than the newspaper stock $x$, then the news vendor sells the stock $x$ giving a revenue of €$(2 - 1)x$. The question is: what is the best order quantity, i.e. what is the optimal value $x^*$? The classic approach is to find the value for

$x$ that maximises the *expected* revenues: Let

$$u(x,v) = \begin{cases} 2v - x & if \quad v \leq x \\ x & if \quad v > x \end{cases} \tag{1.1}$$

be the revenue given order quantity $x$ and sales $v$. The expected value of $u(x, \boldsymbol{v})$ is:

$$E\left[u(x,\boldsymbol{v})\right] = \sum_{v=100}^{200} u(x,v)\tfrac{1}{101} = \sum_{v=100}^{x} \frac{2v-x}{101} + \sum_{v=x+1}^{200} \frac{x}{101} = -\tfrac{1}{101}x^2 + \tfrac{300}{101}x - \frac{9900}{101} \tag{1.2}$$

with optimal order quantity

$$x^* = \arg\max_x E\left[u(x,\boldsymbol{v})\right] = 150$$

with optimal value $E\left[u(x^*,\boldsymbol{v})\right] \approx 124.75$. The best choice for $x$, following the classic approach, is to order 150 newspapers, which will give an optimal revenue of at least €124.- in the long run.

Consider the news vendor ordering 150 newspapers. Let us investigate the Robustness of such a decision. The stock of 150 newspapers implies an investment of €150.-. To make at least €124.- profit, means that at least 137 newspapers have to be sold, based on $2v - 150 \geq 124$. The probability distribution of the random sales $\boldsymbol{v}$ implies that $\Pr\left\{\boldsymbol{v} \geq 137\right\} = \frac{64}{101}$. Interestingly, there exists a more robust order quantity: if the goal is to make at least €124.- profit, then the order quantity $x = 124$ is more robust than $x = 137$, since $\Pr\left\{\boldsymbol{v} \geq 124\right\} = \frac{77}{101} > \frac{64}{101}$. The Robustness of making a profit greater than $\gamma$, given the order quantity $x$ can be expressed more generally. If $\gamma > x$, then the profit target is impossible to reach. Consequently, if $\gamma > x$, then the Robustness is 0. For $100 \leq \gamma \leq x \leq 200$, $2v - x \geq \gamma \Rightarrow v \geq \frac{x+\gamma}{2}$. Substitution gives $\Pr\left\{\boldsymbol{v} \geq \frac{x+\gamma}{2}\right\} = \frac{201 - \frac{x+\gamma}{2}}{101}$. For any profit goal $100 \leq \gamma \leq 200$ and order quantity $100 \leq x \leq 200$, the Robustness of making a profit of at least $\gamma$ given the order quantity $x$ is:

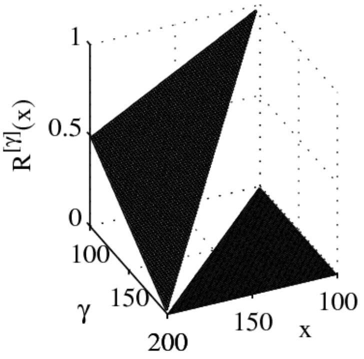


Figure 1.1: Profit Robustness in News Vendor Problem; order quantity $x$, profit target $\gamma$

$$R^{[\gamma]}(x) = \Pr\left\{\gamma \leq u(x,\boldsymbol{v})\right\} = \begin{cases} 0 & \text{if} \quad \gamma > x \\ \frac{201 - \frac{x+\gamma}{2}}{101} & \text{if} \quad \gamma \leq x \end{cases} \tag{1.3}$$

Figure 1.1 is a mesh-plot of $R^{[\gamma]}(x)$. Let $x^{[\gamma]*} = \arg\max_{x} R^{[\gamma]}(x)$ be the most robust order quantity for a given profit target $\gamma$. For fixed $\gamma$, the function $R^{[\gamma]}(x)$ is strictly increasing proportional to $x$ for $x \leq \gamma$. Consequently, the optimal robust order quantity for fixed $\gamma$ is $x^{[\gamma]*} = \gamma$, which can be represented by points with coordinates $\big(\gamma, x, R^{[\gamma]}(x)\big)$ on the line through $(200, 200, \frac{1}{101})$ and $(100, 100, 1)$.

The classic optimal expected value approach gives $x^* = 150$ as the optimal order quantity. The Robustness Programming approach, does not give one single optimal order quantity as a solution. Instead the Robustness Programming Set

$$\mathbb{P} = \left\{ \left( \begin{array}{c} \gamma \\ x^* \\ R^* \end{array} \right) \middle| \, 100 \leq \gamma \leq 200, x^* = \gamma, R^* = \frac{201 - \frac{x+\gamma}{2}}{101} \right\}$$

represents the optimal solution. It it is left to the decision maker, to choose a point from this set.

Robustness Programming is an alternative for the Stochastic Programming approach of maximising expected value. Both approaches serve a different purpose:

- If the news vendor wants to decide about a constant daily order quantity $x$ for the coming year, such that the expected *average* revenues are maximised, then the Stochastic Programming approach of maximising an expected value, is the appropriate way of modelling

- If the news vendor wants to decide about tomorrow's order quantity $x$ with guarantees about tomorrow's revenues, then the Robustness Programming approach is a good alternative.

In this illustration, $R^{[\gamma]*}$ has an analytical solution. The cases studied in Chapter 5 are more complex and no analytical solution for $R^{[\gamma]*}$ is available. The next illustration deals with the Mixture Design Problem that is elaborated in more detail in Section 5.4. The computation of the Optimal Robustness $R^{[\gamma]*}$ is based on an iterative optimisation method using estimates for $R^{[\gamma]}(x)$.

### 1.4.2 Mixture Design illustration

As an industrial illustration consider the following case that is worked out in greater detail in Section 5.4. The case deals with the production of two products. There are eleven Raw Materials (RM) used for the manufacturing of these two products. Over the past weeks the price of each RM varied and it is fair to assume that the RM prices in future weeks have considerable uncertainty. The RM prices determine the cost price of the two products. This illustration is about the Robustness of the total cost price of the two products.

Besides the cost price aspect, it is relevant that the taste, structure and appearance of the two products are according product specifications. Part of the design objective is to find a mixture of the RMs such that the specifications are fulfilled. By comparison, if one wants to make a cake, then not just any combination of flower, butter, sugar, eggs and milk will lead to a cake; only specific proportions of these ingredients will give a satisfactory result. However, there is some flexibility in choosing these ingredient proportions.

The proportion of each of the 11 RMs in the 2 products is a factor that can be controlled. These controllable factors are modelled by variable $X_{i,k}$ representing the proportion of RM $i = 1, .., 11$ in product $k = 1, 2$. The set of feasible mixture designs $\mathbb{F}$ contains all mixture designs $X$ that fulfil the specifications.

Currently, the products are produced according to product design $C \in \mathbb{F}$, which has been used for the past 178 weeks. The cost price of the two products changed considerably over the past 178 weeks. The objective is to find an alternative product design in $\mathbb{F}$, with a high Robustness to save on the current cost price.

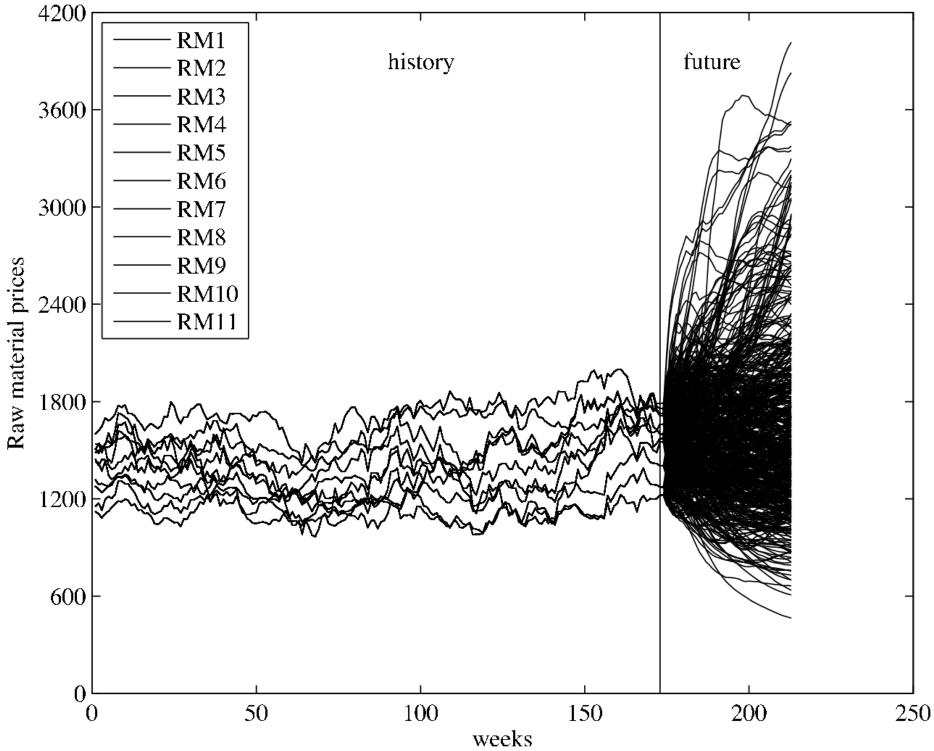


Figure 1.2: Price history of 11 RM until week 178 and future price scenarios from week 179 until 218

The goal is split into two: the saving to be reached and the time span before which to reach the saving. In notation: the absolute savings goal is $\gamma_1$ and the time span to reach the saving is $\gamma_2$. In this illustration, the study is limited to three saving goals of respectively €100,000, €150,000 and €200,000. The timing aspect in the goal (i.e. the second element $\gamma_2$) has the following relevance: If the goal is to save €100,000 in the coming four weeks, then on average €25,000 should be saved per week; if the goal is to save €100,000 in the coming ten weeks, then on average €10,000 should be saved per week and so on. Figure 1.2 illustrates the price history of 11 RMs for the past 178 weeks and 20 samples per RM for the future week prices between week 179 and 218.

Let $\boldsymbol{v}$ be a random $11 \times 40$ matrix, corresponding to the random future prices per ton for each RM for the weeks 179 until 218. Samples are generated with a Geometric Brownian Motion time series model[1]. The weekly production volume of product $k$ is $q_k$ (in 1000Kg units) for $k = 1, 2$. The current week is 178. Let $\tau = 1$ correspond to the first future week, namely 179; $\tau = 2$ corresponds to future week 180; etcetera. The total saving over future weeks until $\gamma_2$, for a mixture design $X \in \mathbb{F}$ with respect to the mixture design $C \in \mathbb{F}$ currently used for production, is

$$u^{[\gamma_2]}(x, v) = \sum_{i=1}^{11} \sum_{k=1}^{2} q_k \left(\mathrm{C}_{i,k} - X_{i,k}\right) \sum_{\tau=1}^{\gamma_2} v_{\tau,i} \tag{1.4}$$

with future RM prices $v \in \mathbb{V}$. The saving model (1.4) determines the Robustness of reaching savings goal $\gamma_1$ within the period up to week goal $\gamma_2$:

$$R^{[\gamma]}(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}^{[\gamma]}(x)\right\}, \text{ where } \mathbb{H}^{[\gamma]}(x) = \left\{v \in \mathbb{V} | \gamma_1 \leq u^{[\gamma_2]}(x, v)\right\} \tag{1.5}$$

The corresponding Robustness Programming Set (RPS) is

$$\mathbb{P} = \left\{ \left. \begin{pmatrix} \gamma \\ X^* \\ R^* \end{pmatrix} \right| R^* = \max_{X \in \mathbb{F}} R^{[\gamma]}(X), R^* = R^{[\gamma]}\left(X^*\right) \right\} \tag{1.6}$$

Finding elements in $\mathbb{P}$ in an analytical way is not possible if an analytical solution for the optimisation problem in (1.6) is not available. This is often the case in practice and

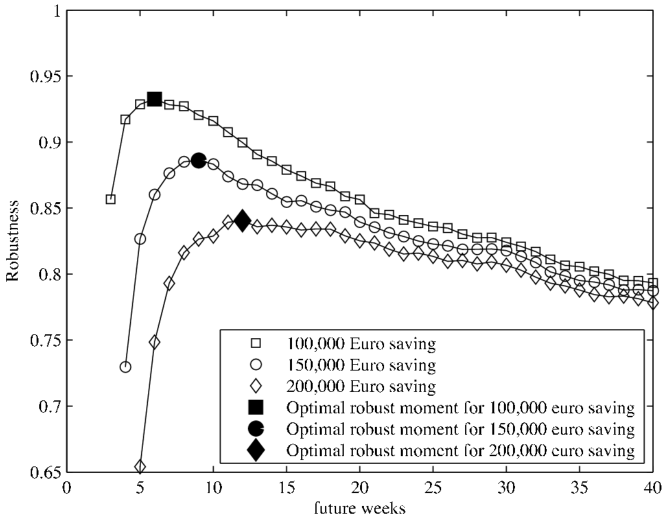


Figure 1.3: Optimal Robustness of €100,000, €150,000 and €200,000 saving respectively, during the period of *future weeks* up to $\gamma_2$

[1] The choice for this model is motivated in Section 5.4

therefore the approach is to use Robustness Programming methods presented in this thesis to estimate $R^{[\gamma]}(x)$ and use an iterative solver to find optimal values for $X^*$ and $R^*$, given the goals $\gamma_1$ and $\gamma_2$. Figure 1.2 shows an interpolation of the optimal Robustness given a finite number of goal values: $\gamma_1 \in \{100000, 150000, 150000\}$ and $\gamma_2 = 1, .., 40$.

The resulting curves in Figure 1.3 can be explained as follows: All events leading to a saving of €200,000, also lead to a saving of at least €150,000, therefore the Robustness curve of the €200,000 goal is below the Robustness curve of the €150,000 goal. The same holds for the €150,000 saving goal with respect to the €100,000 saving goal. The three graphs have a unimodal shape. An explanation is the following. It has been argued that for increasing $\gamma_2$ the average saving goal per week goes down. On the other hand, for increasing $\gamma_2$ the price uncertainty goes up. This is illustrated by Figure 1.2, that shows increasing variation of the simulated future prices. Apparently, the negative effect of increased price uncertainty becomes bigger than the positive effect of the lower weekly saving goals at some point in time.

The idea is that the RPS and Figure 1.3 can be used to support decision making. The curves in the figure suggest that if for instance the goal is to save €150,000, then it is most robust to aim at reaching the saving goal 9 weeks from now. On the other hand, the lower saving goal of €100,000 can be reached with a higher Robustness and within less weeks, i.e. already in week 6. There is no mathematical basis to argue which of these two options is best. It is left to the preference of the decision maker to come to a decision. The idea is that Figure 1.3 in combination with the RPS, gives relevant information to support such decision making. For instance, if the decision maker chooses the €150.000 saving goal, then the RPS can be queried to find the corresponding best mixture design $X^*$ for $\gamma_1 = 150000, \gamma_2 = 9$.

## 1.5 Robustness Programming literature

In retrospective, it is hard to define unambiguously when Robustness was considered as a design objective for the first time. The ancient Egyptians about 3500 B.C. did understand the concept of *chance* Ghahramani (2000) and were capable of constructing very robust pyramids.

A recurring objective in finance, is to find good investment combinations that carry low risk. A low risk portfolio, can be interpreted as a robust portfolio. In a paper by Mark Rubinstein, titled *Markowitz's "Portfolio Selection": A Fifty−year Retrospective* (Rubinstein, 2002), an overview is presented, of historic important documents in the financial field. Markowitz (1999) pointed out, that *The merchant of Venice*, Act 1, Scene 1, by William Shakespeare (Shakespeare, 1914), contains a dialogue which can be seen as an early awareness that diversification of investments reduces risk: "...I Thank my fortune for it, my ventures are not in one bottom trusted. Nor to one place; nor is my whole estate upon the fortune of this present year...". Mark Rubinstein also refers to Bernoulli's St. Petersburg paradox article (Bernoulli, 1738), in which it is argued that risk-averse investors will want to diversify: "...it is advisable to divide goods which are exposed to some small danger into several portions rather than to risk them all together".

The 18th century contemporaries Bernoulli, Laplace and Bayes, are often considered as the founders of Probability Theory. An often addressed topic, was the problem of making the best decision under uncertainty, based on a probabilistic argument. One can argue that the best decision under uncertainty, is the most robust decision and can therefore be seen as an early example of Robustness Programming.

In 1952, Harry Markowitz published his landmark paper "Portfolio Selection" (Markowitz, 1952), which is now perceived as the moment of birth of modern Financial Economics. Markowitz suggested to optimise the expected value of a portfolio, which is corrected for a fixed proportion -say a factor $k$- of variance. For fixed $k$, portfolios connected to a high variance are penalised more than low variance portfolios. Increasing the factor $k$ results in more robust portfolio selections.

In parallel to the scientific progress in the financial field, also industries developed techniques for designing robust products. Illustrations related to early attempts of robust product design, are for instance presented in the thesis of Freek Huele (1998) and deal with design efforts to reduce the negative effect of variations in raw materials, in the context of Guinness beer production, that dates back to the beginning of the $20^{th}$ century.

"During the Second World War, British military leaders asked scientists and engineers to analyse several military problems: the deployment of radar and the management of convoy, bombing, antisubmarine and mining operations. The application of mathematics and the scientific method to military operations was called Operations Research" (Winston, 1993). In 1947, George Dantzig was one of the first mathematicians who translated the lessons learned, into his famous scientific contributions: the framework of Linear Programming and the Simplex Method (Dantzig, 1951). In 1955, George Dantzig wrote another famous paper, titled "Linear Programming under Uncertainty" (Dantzig, 1955), which is often considered to mark the beginning of the structural development of the mathematical science area known now as Stochastic Programming. As the name implies, the framework of Stochastic Programming is based on the assumption that uncertain factors can be modelled as random vectors. The works of Dantzig and Markowitz in the mid-fifties can be seen as one of the first fundamental methods for making best decisions under uncertainty. The wide variety of "Numerical Techniques for Stochastic optimization" elaborated on by Ermoliev and Wets (1988), show the scientific progress of Stochastic Programming in the three decades that followed the early works of Dantzig and Markovitz.

In the 1980's Genichi Taguchi, an industrial director with an engineering degree and statistical background, introduced one of the first methods for robust design which got widely accepted by industry (Taguchi, 1986). In the Taguchi approach, quality-loss is expressed in a monetary value, such that the quality aspect can be included in the overall cost structure of a product, which can be optimised by selecting the experiment with the best results. Possibly, one of the reasons why manufacturing industry embraced the Taguchi method, was that it is relatively simple to apply and interpret. Birge and Louveaux (1997, page 37) claim that "with hindsight Taguchi methods can be seen as examples of Stochastic Programming, although they are often not described this way".

Since the 1980's the Stochastic Programming and industrial engineering world continued to evolve rapidly where the interaction between mathematics, statistics and engineering resulted in a multi-form of new concepts, such as quality engineering, safety engineering and structural reliability engineering. In the early 1990's, Otto and Antonsson (1993) and Parkinson et al. (1993), extended Taguchi's Robust Design concepts to

Robust Design Optimisation (RDO), by adding NonLinear Programming (NLP) to the framework. In RDO, a distinction is made between two types of Robustness: objective Robustness and constraint Robustness.

Along with the more sophisticated modelling power of RDO, came the problem of how to solve the NLP problems. In more recent years, for instance Hendrix (1998) studied the application of global optimisation techniques for RDO problems.

The possible computational intractability of RDO problems and large Stochastic Programming problems in general, motivated Ben Tal and Nemirovski to study the potential of a new collection of techniques called Semi-Definite Optimisation (SDO), for optimisation problems under uncertainty. Ben-Tal and Nemirovski (2002) developed the concept of Robust Optimisation (RO), where uncertainty is modelled via a so-called *uncertainty set*: uncertain variates can only have realisations in the uncertainty set. Robust optimisation deals with finding an optimal design, which is feasible for all elements of the uncertainty set. The modelling of the uncertainty set can be based on the assumption that uncertain variates are stochastic, such that the probability mass of the uncertainty set approaches 1. Conversely, the uncertainty set can also be modelled by deductive reasoning using correct analytical models. For example, an electronic component such as a resistor is known to have some resistance within tolerance limits. Suppose that only resistors that pass the tolerance limit check are put on the market. In this situation, one can argue that the tolerance limits provide sufficient information to construct the uncertainty set and one does not need a stochastic model to describe the variation in resistance.

In the context of modern robust supply chain design, it is interesting to observe that Kleijnen et al. (2003) is studying Taguchi-style approaches which originates from product engineering. Initially, the Robust Optimisation concept developed by Ben-Tal and Nemirovski, was applied in the field of product engineering. More recently, Goldfarb and Iyengar (2003) showed very powerful applications in Mathematical Finance of the RO framework in the context of Robust Portfolio Optimisation.

## 1.6 Research Outline

Robustness Programming (RP) is based on Stochastic Programming and in particular on the sub-domain called Probabilistic Programming. This way, the foundation of Stochastic Programming, i.e. Mathematical Programming, Statistics and Probability Theory, is indirectly also the foundation of RP. The intention of the RP framework is to cover a wide scope of mathematical concepts which are useful for RP. In principle, many methods from Statistics and Stochastic Programming are useful for RP and are a source of inspiration for this research. However, these methods are often problem specific or have bad Robustness optimisation characteristics. On the one hand, there is a need for generally applicable RP methods. On the other hand, existing RP methods can be generalised and improved. In the sequel, the research objectives are given, which imply the research questions and research approach.

### Research objectives

1. The first objective is to design an RP framework, for expressing RP problems and defining RP methods in a uniform and systematic notation
2. The second objective is to determine conditions that identify applicable RP methods, when given specific RP problem properties
3. The third objective is to improve and generalise existing Robustness Programming methods and design new Robustness Programming methods, that estimate Robustness effectively and efficiently
4. The fourth objective is to assess the quality of the Robustness Programming methods, from a theoretical and empirical point of view

### Research questions

1. What are the generic components in an RP problem, that enable both a systematic RP problem notation as well as defining RP methods using the same notation?
2. What are the characteristics of an RP problem, that provide sufficient information to decide about applicable RP solution methods?
3. Which mathematical properties of Robustness Programming methods give information about its quality, relevant for Robustness estimation and Robustness optimisation?
4. How can the performance of Robustness Programming methods be compared?

### Research approach

The research is triggered by case studies and is done in a cyclic form, where theory and practice alternate in the following steps: (1) analyse case studies; (2) select existing or modified RP methods or design new RP methods applicable to case studies; (3) apply RP methods in case studies; (4) assess results; (5) analyse theoretical properties of RP methods and explain results; (6) find ways to improve RP methods or use the results as inspiration to find new RP methods and go back to step 2. In this respect, one case from Environmental Economics and two industrial cases from Unilever are studied. The following research approach is used to answer the corresponding research questions.

1. The RP framework that evolved during the development of RP methods for solving practical cases is the output of the research. In this thesis, the framework is presented first. The applicability of the Robustness Programming Framework is shown by basing all definitions, theorems and conclusions in subsequent chapters on this framework.
2. Conditions that determine which RP methods are applicable, given RP problem properties, are called *sufficiency conditions for applicability*. Determination of these conditions is based on the following approach. RP methods are designed, both for

specific RP problems as well as for general RP problems. This way, there exists an RP method for every RP problem that fits the RP framework. For each RP method, the required mathematical properties of the RP problem are investigated.

3. The approach to study the quality of RP methods, is to define and investigate the efficiency and effectiveness of RP methods and to assess the applicability of RP methods. The Monte Carlo (MC) Robustness estimation method is used as a reference method, because the MC method is commonly used and generally applicable for Robustness estimation. The approach is to compare the efficiency and effectiveness of new RP methods and improved RP methods with the MC method.

4. The approach is to design indicators for the performance of RP methods, in such a way that similar results are obtained when experiments are repeated. The approach is to be as independent as possible, with respect to computer speed and software implementation.

### Practical aspects and research approach

The case studies involve software implementations of the RP methods. At the moment of writing, there does not exist standard software for modelling and solving large-scale nonlinear Robustness Programming problems[2]. Experimental software implementations have been carried out for Robustness estimation and optimisation, using existing NLP software to solve Robustness optimisation problems.

The cases that are discussed in Chapter 5 were provided in Matlab® coding. This led to the practical choice to use Matlab® and the standard Matlab Optimisation Toolbox® for the validation of the experimental Robustness estimation implementations. The studied RP problems are nonlinear constrained problems and are solved using the standard FMINCON solver (see Venkataraman, 2002). The FMINCON solver uses a *Sequential Quadratic Programming* (SQP) method in combination with *Line Search* methods[3]. The consequence of this practical context, is that Robustness optimisation is only validated using an SQP method. Robustness optimisation using any other method (such as sequential linear programming, random search or an evolutionary algorithm) is left for future research (see Section 6.5).

### Organisation of the thesis

After the introduction of Robustness Programming in Chapter 1, the definitions, notations and basic models of the Robustness Programming Framework are introduced in Chapter 2. In Chapter 3, methods for estimating Robustness via sampling and bounding are described. In Chapter 4, computation techniques for improving the efficiency and effectiveness of Robustness Programming methods are discussed and performance indicators

[2] For stochastic *linear* programming with chance constraints, a software package called SLP-IOR (Kall and Mayer, 1996) is available.

[3] Although SQP methods were originally designed for continuous and (twice) differentiable functions (Bazaraa et al., 1993), later studies revealed that SQP is relatively stable even for non-differentiable and noisy functions (see for instance Schittkowski, 1985, 1994, 2005).

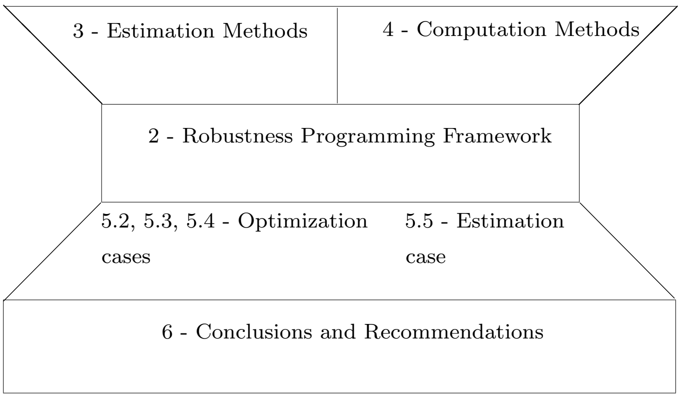

3 - Estimation Methods
4 - Computation Methods
2 - Robustness Programming Framework
5.2, 5.3, 5.4 - Optimization cases
5.5 - Estimation case
6 - Conclusions and Recommendations

Figure 1.4: Thesis Outline

are given. In Chapter 5, the case studies are discussed. They deal with the application of methods for Robustness Programming. Finally, conclusions and recommendations for future research are given in Chapter 6.

The structure of the thesis is sketched in figure 1.4. The Robustness Programming Framework is depicted in the centre. On the theoretical side (top), there are solution methods for problems that fit the framework. On the practical side (bottom), there are problems, which can be modelled according the framework and solved by the RP methods. In subsequent chapters, a miniature version of Figure 1.4 is given on the top of each even page, to indicate (in black) which part of the big picture is being discussed.

# Chapter 2
# Robustness Programming Framework

Robustness of an object is the probability that uncontrollable factors take realisations such that the object has properties as required. An optimal Robustness design is a plan for the creation of an object, with maximum probability of having object properties as required. In this chapter, the framework is introduced for modelling and optimising Robustness.

## 2.1 The concepts

The focus of Robustness Programming (RP) is on estimation, computation and optimisation of Robustness of object properties that depend on uncontrollable factors. These object properties are called uncertain object properties. Object properties that do not depend on uncertain factors are called deterministic object properties. The following notation and models are the core of the Robustness Programming Framework.

- J is the number of deterministic object properties
- S is the number of uncertain object properties
- I is the number of controllable factors
- N is the number of uncontrollable factors
- Controllable factors are modelled as a vector $x \in \mathbb{R}^{\mathrm{I}}$
- Uncontrollable factors are modelled as a random vector $\boldsymbol{v}$ defined on a probability space $(\mathbb{V}, \mathcal{V}, \mathrm{Pr}_{\boldsymbol{v}})$, with realisations $v \in \mathbb{V} \subseteq \mathbb{R}^{\mathrm{N}}$
- Deterministic object properties depend on controllable factors $(x)$ and are modelled by a function $d : \mathbb{R}^{\mathrm{I}} \longrightarrow \mathbb{R}^{\mathrm{J}}$, where $d(x)$ is a continuous function of $x$
- Uncertain object properties depend on controllable factors $x$ and uncontrollable factors $v$ and are modelled by a function $u : \mathbb{R}^{\mathrm{I}} \times \mathbb{R}^{\mathrm{N}} \longrightarrow \mathbb{R}^{\mathrm{S}}$ where $u(x, v)$ is a continuous function of $v$. Continuity with respect to $x$ is discussed in Section 2.4
- The requirement for deterministic object property $j$ is modelled as a closed interval $[\Lambda_j, \Upsilon_j]$, with lower bound $\Lambda_j \in \overline{\mathbb{R}}$ and upper bound $\Upsilon_j \in \overline{\mathbb{R}}$ for $j = 1, ..,$J in the extended reals [1]
- The requirement for uncertain object property $s$ is modelled as a closed interval $[\mathrm{L}_s, \mathrm{H}_s]$, with lower bound $\mathrm{L}_s \in \overline{\mathbb{R}}$ and upper bound $\mathrm{H}_s \in \overline{\mathbb{R}}$ for $s = 1, ..,$S
- $\Lambda_j \leq d_j(x) \leq \Upsilon_j$, $j = 1, .., \mathrm{J}$ are called *deterministic restrictions*
- $\mathrm{L}_s \leq u_s(x, v) \leq \mathrm{H}_s$, $s = 1, .., \mathrm{S}$ are called *uncertain restrictions*

[1] The set $\overline{\mathbb{R}} = \{-\infty\} \cup \mathbb{R} \cup \{\infty\}$ is called the set of extended reals.

Consider a food product design, where $x_i$ represents the proportion of raw material $i$ in a mix of I raw materials. Suppose there is 1 (=J) deterministic object property. Let $x_8$ represent the quantity of orange juice in the food product and the product should at least contain 30% orange juice. A translation of this requirement, is the following deterministic restriction:

$$\Lambda_1 = 0.3 \leq d_1(x) = \frac{x_8}{\sum\limits_{i=1}^{\mathrm{I}} x_i} \leq \infty = \Upsilon_1 \tag{2.1}$$

For the general case, all designs that satisfy the deterministic restrictions can be collected in a set. The set of *deterministic feasible designs* is defined as

$$\mathbb{X} = \{x \in \mathbb{R}^{\mathrm{I}} | \Lambda_j \leq d_j(x) \leq \Upsilon_j,\ j = 1, .., \mathrm{J}\} \tag{2.2}$$

The concentration of vitamin C in orange juice can vary, depending on the country of origin, weather conditions during growth and harvest period and storage time. Likewise, there is uncertainty in the vitamin C concentration of the other raw materials. Let $v_i$ model the uncertain vitamin C concentration in raw material $i$. The following uncertain restriction models the requirement that the food product contains at least 0.05% vitamin C:

$$\mathrm{L}_1 = 0.0005 \leq u_1(x, v) \leq \infty = \mathrm{H}_1 \tag{2.3}$$

with $u_1(x, v) = \frac{v^\intercal x}{\sum\limits_{i=1}^{\mathrm{I}} x_i}$. As $\boldsymbol{v}$ is a random vector, the value of $v$ is unknown a priori. Consequently, it may be uncertain for chosen composition $x$, whether or not condition (2.3) is fulfilled.

## 2.2 Robustness

The random vector $\boldsymbol{v}$ defined on probability space $(\mathbb{V}, \mathcal{V}, \mathrm{Pr}_{\boldsymbol{v}})$ can be used to make explicit statements about the *probability* of satisfying all uncertain restrictions.

**Definition 2.1** *The Distribution function* $F : \mathbb{R}^{\mathrm{N}} \longrightarrow [0, 1]$*, induced by probability* $\mathrm{Pr}_{\boldsymbol{v}}$ *on* $(\mathbb{V}, \mathcal{V})$ *is the function*

$$F(v) = \mathrm{Pr}\left\{\boldsymbol{v}_n \leq v_n \textit{ for } n = 1, .., \mathrm{N}\right\}$$

A central role is played by the subset of $\mathbb{V}$ for which the object properties are as required. This leads to the definition of the so-called Happy set :

**Definition 2.2** *The* Happy set $\mathbb{H}(x)$ *is defined, as the set of realisations of random vector* $\boldsymbol{v}$ *which satisfy all uncertain restrictions:*

$$\mathbb{H}(x) = \{v \in \mathbb{V} | \mathrm{L}_s \leq u_s(x, v) \leq \mathrm{H}_s,\ s = 1, .., \mathrm{S}\}$$

The set $\mathbb{H}(x)$ is called the *Happy set*, since elements of this set correspond to the favourable situation that the S uncertain properties $u_s(x, v)$ of the object under consideration are as intended, i.e. these properties are as required between the bounds $\mathrm{L}_s$ and $\mathrm{H}_s$ for $s = 1, ..,$S.

**Corollary 2.1** *The Happy set belongs to the class of Borel measurable sets*

$$\mathbb{H}(x) \in \mathcal{B}\left(\mathbb{R}^{\mathrm{N}}\right)$$

*following from the definition by (Randolph, 1968, page 245, definition 2), since the Happy set is a* closed *subset of* $\mathbb{R}^{\mathrm{N}}$ *for given* $x$.

**Definition 2.3** *A design* $x \in \mathbb{X}$ *is called* robust*, if and only if the design fulfils all requirements for* all *possible realisations in* $\mathbb{V}$*:*

$$\mathbb{H}(x) = \mathbb{V} \Longleftrightarrow \text{"}x \textit{ is robust"}$$

Consequently, we will say a design has *some* level of *Robustness*, if the design fulfils all requirements for a *subset* $\mathbb{S} \subset \mathbb{V}$ of all possible realisations of $v$:

$$\mathbb{H}(x) \subset \mathbb{V} \Longleftrightarrow \text{"}x \text{ is } \textit{less than} \text{ robust"}$$

Conceptually, the level of Robustness $R(x)$ is *some* measure that depends on the design $x$. Definition 2.4 gives the Robustness measure investigated in this thesis.

**Definition 2.4** *The Robustness is the probability of satisfying all uncertain restrictions:*

$$R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} \tag{2.4}$$

A more explicit expression for Robustness is obtained as follows. Let $I : \mathbb{R}^{\mathrm{N}} \times \mathbb{R}^{\mathrm{I}} \longrightarrow \{0, 1\}$ be the Happy set indicator function, where

**Definition 2.5**

$$I(v, x) = \begin{cases} 1 & \textit{if } v \in \mathbb{H}(x) \\ 0 & \textit{otherwise} \end{cases} \tag{2.5}$$

This function *indicates* whether (1) or not (0) a realisation $v$ of random vector $\boldsymbol{v}$ is in the Happy set. It can be shown that $E_{\boldsymbol{v}}[I(\boldsymbol{v}, x)] = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} = R(x)$ (based on the arguments of Jacod and Protter, 2004; Robert and Casella, 1999). The Robustness, respectively related to *discrete* distributions and *continuous* distributions, is defined as follows.

**Corollary 2.2** *Let* $\boldsymbol{v}$ *be a discrete random vector defined on Probability Space* $(\mathbb{V}, \mathcal{V}, \Pr_{\boldsymbol{v}})$*, with countable set* $\mathbb{V}$*, then:*

$$R(x) = \sum_{v \in \mathbb{V}} I(x, v) \Pr\{\boldsymbol{v} = v\} \tag{2.6}$$

**Corollary 2.3** *Let $\boldsymbol{v}$ be a continuous random vector on Probability Space* $(\mathbb{V}, \mathcal{V}, \Pr_{\boldsymbol{v}})$, *defined by a probability density function* $f : \mathbb{R}^{\mathrm{N}} \to \mathbb{R}_+$, *then:*

$$R(x) = \int\limits_{\mathbb{R}^{\mathrm{N}}} I(x, v) f(v) dv \tag{2.7}$$

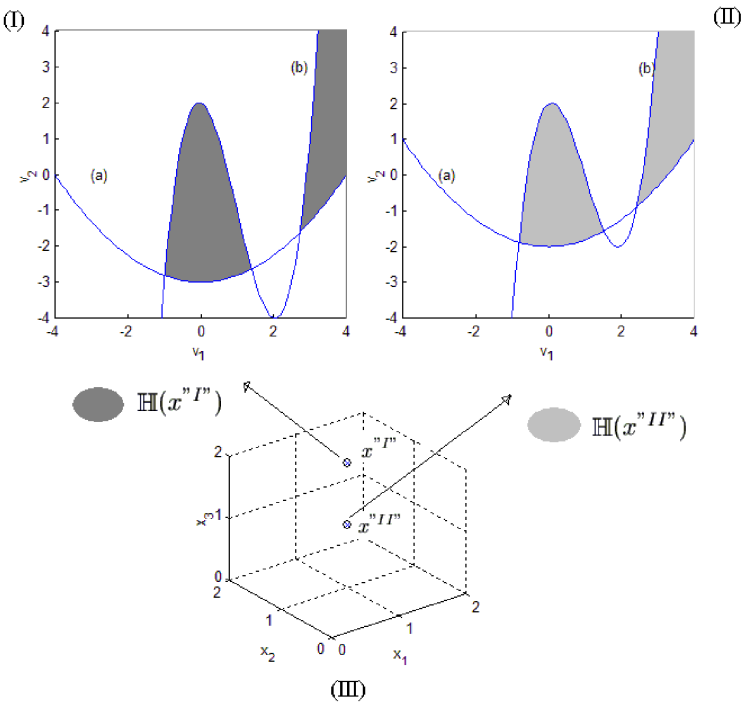


Figure 2.1: Examples of $\mathbb{H}(x)$

**Example 2.1** *Two uncertain restrictions are given as depicted by Figure 2.1*

$$\mathrm{L}_1 = 0 \leq u_1(x, v) = -(x_1 - \frac{13}{16})v_1^2 - (x_2 - 1)v_1 + v_2 + 1 + x_3 \leq \infty = \mathrm{H}_1$$
$$\mathrm{L}_2 = 0 \leq u_2(x, v) = (x_1 + \frac{1}{3})v_1^3 + (x_2 - 5)v_1^2 + (1\frac{2}{3} - x_3)v_1 + 2 - v_2 \leq \infty = \mathrm{H}_2$$

*Figure 2.1 depicts the Happy set for two designs,* $x^{[1]} = [1, 1, 2]^{\intercal}$ *and* $x^{[2]} = [1, 1, 1]^{\intercal}$. *The area of the Happy set are enclosed by curves defined by the restrictions, i.e. for* $x^{[1]}$ *the curves in (I) correspond to*

$$u_1(x^{[1]}, v) = -\frac{3}{16}v_1^2 + v_2 + 3 = 0 \qquad (a)$$
$$u_2(x^{[1]}, v) = 1\frac{1}{3}v_1^3 - 4v_1^2 - \frac{1}{3}v_1 + 2 - v_2 = 0 \qquad (b)$$

*and for* $x^{[2]}$ *the curves (II) are given by*

$$u_1(x^{[2]}, v) = -\frac{3}{16}v_1^2 + v_2 + 2 = 0 \qquad (a)$$

$$u_2(x^{[2]}, v) = 1\frac{1}{3}v_1^3 - 4v_1^2 + \frac{2}{3}v_1 + 2 - v_2 = 0 \qquad (b)$$

$\mathbb{H}(x^{[1]})$ *and* $\mathbb{H}(x^{[2]})$ *are the grey areas between curves (a) and (b)*

*The Robustness* $R(x^{[1]}) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}(x^{[1]})\right\}$ *and* $R(x^{[2]}) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}(x^{[2]})\right\}$*, depend on the definition of* $(\mathbb{V}, \mathcal{V}, \Pr_{\boldsymbol{v}})$ *and can define a non-trivial integral to compute. If* $\boldsymbol{v}$ *follows a uniform distribution on* $[-4, 4] \times [-4, 4]$*, then computing the surface between the curves (a) and (b) via integration gives that* $0,1904 = R(x^{[1]}) > R(x^{[2]}) = 0.1653$.

## 2.3 Robustness and optimisation

The main aim of Robustness Programming (RP) is to find designs $x^* \in \mathbb{X}$ corresponding to the maximum Robustness. A typical Robustness Programming problem is the following:

$$R^* = \max_{x \in \mathbb{X}} \left[\Pr\left\{\boldsymbol{v} \in \mathbb{H}(x)\right\}\right] \tag{2.8}$$

**Example 2.2** *We give an RP problem that by exception has an analytical solution for* (2.8)*. Let random (univariate) variable* $\boldsymbol{v}$ *follow an Exponential distribution with parameter* $\lambda > 0$*, i.e. the Probability Density Function (PDF)* $f(v) = \begin{cases} \lambda e^{-\lambda v}, & v \geq 0 \\ 0, & v < 0 \end{cases}$ *and Distribution Function* $F(v) = \begin{cases} 1 - e^{-\lambda v}, & v \geq 0 \\ 0, & v < 0 \end{cases}$.

*Let* $\mathbb{X} = \mathbb{R}_+$, $u_1(x, v) = \alpha x - v$ *and* $u_2(x, v) = v - \beta x$ *with* $\alpha < \beta$.

*Define* $\mathbb{H}(x) = \{v \in \mathbb{R}_+ \,| u_1(x, v) \leq 0,\ u_2(x, v) \leq 0\} = \{v \in \mathbb{R}_+ \,| \alpha x \leq v \leq \beta x\}$.

*The Robustness can be computed analytically by*

$$R(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}(x)\right\} = 1 - e^{-\beta\lambda x} - \left(1 - e^{-\alpha\lambda x}\right) = e^{-\alpha\lambda x} - e^{-\beta\lambda x} \tag{2.9}$$

*Notice that* $R(0) = 0$, $R(x) > 0$ *for* $x > 0$ *and* $\lim_{x \to \infty} R(x) = 0$*. It can be shown that* $x^* = \frac{\ln(\alpha\lambda) - \ln(\beta\lambda)}{\alpha\lambda - \beta\lambda}$ *is a unique solution for* $\frac{dR(x)}{dx} = 0$*. Consequently,* $R(x)$ *is an unimodal function with global maximum* $R^* = R(x^*)$.

**Example 2.3** *In Example 2.1 we take*

$$\begin{aligned} \mathbb{X} = \{x | x_1 = 1, x_2 = 1, x_3 \in [1, 4]\} \\ \boldsymbol{v}_i \sim N(0, 1) \textit{ for } i = 1, 2 \end{aligned} \tag{2.10}$$

*In contrast to Example 2.2,* (2.8) *is not solved analytically. Instead,* $R^*$ *is determined using an iterative numerical optimisation algorithm, where* $R(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}(x)\right\}$ *is estimated for given* $x$ *with methods that are explained in Chapters 3 and Chapter 4. The set* $\mathbb{X}$ *is depicted in the left graph of Figure 2.2. In the right graph, the Robustness is depicted as a function of* $x_3$*. The numerically found optimal Robustness design* $x^* = [1, 1, 3.2]^\intercal$ *corresponds to the maximum Robustness* $R^* = 0.65$.

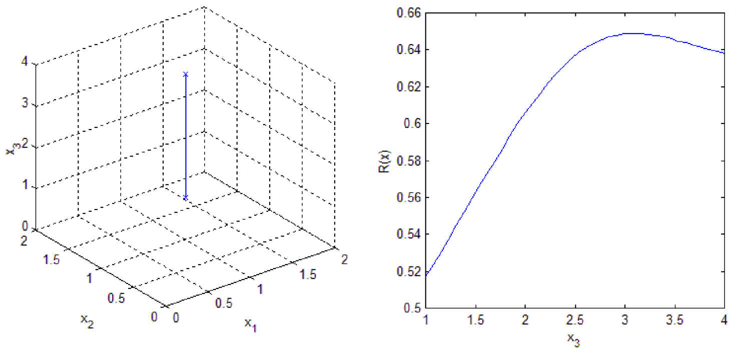


Figure 2.2: Examples of Robustness optimisation

The News Vendor Problem, as introduced in Section 1.4.1 and the Mixture Design problem in Section 1.4.2, refer to a relation between profit goals and Robustness. The general idea of the relation between goals and Robustness is included in the framework via a goal parameter $\gamma \in \mathbb{R}^{\mathrm{K}}$. In the News Vendor Problem, the goal parameter ($\gamma \in \{100, .., 200\}$) defines the lower bound for uncertain restriction $\gamma \leq u(x, v)$ as in (1.3). In the Mixture Design problem, the goal parameter is two-dimensional. The first element ($\gamma_1 \in \{100000, 150000, 200000\}$) defines the savings target and the second element ($\gamma_2 \in \{1, .., 40\}$ ) the number of weeks for realising the saving. The generalisation of the Robustness and Happy set concept, in relation to the goal parameter idea, is that the goal parameter defines the model components of the Happy set. The parameterised Robustness $R^{[\gamma]}(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}^{[\gamma]}(x)\right\}$ is based on a parameterized Happy set, defined as

$$\mathbb{H}^{[\gamma]}(x) = \{v \in \mathbb{V} | \mathrm{L}^{[\gamma]} \leq u^{[\gamma]}(x, v) \leq \mathrm{H}^{[\gamma]}\}. \tag{2.11}$$

The relationship between goals, optimal decisions and optimal Robustness is modelled as the Robustness Programming Set (RPS) and is defined as:

**Definition 2.6**

$$\mathbb{P} = \left\{ \begin{pmatrix} \gamma \\ x^* \\ R^* \end{pmatrix} \in \begin{pmatrix} \mathbb{R}^{\mathrm{K}} \\ \mathbb{R}^{\mathrm{I}} \\ [0,1] \end{pmatrix} \middle| \begin{array}{c} R^* = \max\limits_{x \in \mathbb{X}} R^{[\gamma]}(x) \\ x^* = \arg\max\limits_{x \in \mathbb{X}} R^{[\gamma]}(x) \end{array} \right\} \tag{2.12}$$

The concept of Robustness Programming has been introduced from a Robustness optimisation perspective, where $R(x)$ is the objective function. The RP framework can be generalised by following the Mathematical Programming framework, where optimisation problems are defined by objective functions and constraint functions. Let $R_1(x)$, $R_2(x)$ and $R_3(x)$ be three different Robustness functions. An example of an RP problem with constraints is

$$\begin{aligned} R^* = & \max R_1(x) \\ & \text{s.t.} \\ & x \in \mathbb{X} \\ & R_2(x) \geq 0.95 \\ & R_3(x) \geq 0.95 \end{aligned}$$

## 2.4 RP framework standardisation

The RP modelling framework is introduced in Section 2.1. The following standardisation is introduced, in order to present methods for estimating (Chapter 3) and computing (Chapter 4) the Robustness (2.4) in a systematic way.

### The uncontrollable factors

**Definition 2.7** Standard uncontrollable factors *are modelled as a random vector* $\boldsymbol{v}$ *defined on the probability space* $(\mathbb{V}, \mathcal{V}, \Pr_{\boldsymbol{v}})$*, where the elements* $\boldsymbol{v}_n$*,* $n = 1, .., \mathrm{N}$ *are independently distributed and the first and second moments of each element are finite.*

**Corollary 2.4** $COV(\boldsymbol{v}_i, \boldsymbol{v}_j) = 0$ *for* $i, j = 1, .., \mathrm{N}$ *and* $i \neq j$*.*

**Corollary 2.5** *The sample space of* $\boldsymbol{v}$ *can be written as a Cartesian product* $\mathbb{V} = \prod_{n=1}^{\mathrm{N}} \mathbb{V}_n$*, with* $\mathbb{V}_n$ *the sample space of random element* $\boldsymbol{v}_n$*.*

The standard Robustness Programming model can be used in the following way. Let the elements of a random vector $\tilde{\boldsymbol{v}}$ be dependent, $E(\tilde{\boldsymbol{v}}_n) \neq 0$ and $VAR(\tilde{\boldsymbol{v}}_n) \neq 1$, $n = 1, .., \mathrm{N}$. Let $\tilde{u}(x, \tilde{v})$ be the corresponding uncertain object properties model. If there exists a mapping $D : \mathbb{V} \longrightarrow \tilde{\mathbb{V}}$ with $\tilde{v} = D(v)$, then the RP framework can be used, since $\tilde{u}(x, D(v)) = u(x, v)$. The idea is that all dependency structures between uncontrollable factors, central displacements ($E(\tilde{\boldsymbol{v}}_n) \neq 0$) and scaling ($VAR(\boldsymbol{v}_n) \neq 1$) are modelled *in* the definition of the uncertain object properties function $u$.

A typical example of mapping $D$ is $\tilde{v} = Tv + \mu$, with $T$ a M×N matrix and $\mu \in \mathbb{R}^{\mathrm{M}}$, which is a linear mapping to define a random vector $\tilde{\boldsymbol{v}}$ with dependent elements, $E(\tilde{\boldsymbol{v}}) = \mu$ and $COV(\tilde{\boldsymbol{v}}_i, \tilde{\boldsymbol{v}}_j) = \Sigma_{i,j}$ for $i, j = 1, .., \mathrm{M}$ and $TT' = \Sigma$.

### The Happy set

The Happy set is based on the model $u(x, v)$ of the uncertain object properties. Two situations can be distinguished:

- $R(x)$ is to be determined for a given value of $x$, or a finite number of values of $x$. Then the continuity of $u(x, v)$ with respect to $x$ is not relevant.

- $R(x)$ is either the objective function or a constraint function in a Robustness optimisation problem. Then the continuity of $u(x, v)$, with respect to $x$, is relevant when using standard NLP methods for optimisation.

## 2.5 Concluding remarks

This chapter introduced in a formal way the ingredients of the Robustness Programming Framework where several examples are given as illustrations. In Chapter 3, Robustness estimation methods are discussed for estimating (2.4) in such a way that (2.8) can be solved using standard NLP algorithms.

# Chapter 3
# Robustness Estimation Methods

## 3.1 Motivation

In Chapter 2, Robustness has been defined as $R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$. From studies in multivariate statistical analysis (such as Narayan, 1996) one can conclude that in many cases closed form expressions for $\Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$ are either computational intractable, or do not exist. For that reason, in this chapter we will pay attention to Robustness Estimation methods.

An often used method to estimate $\Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$ is Monte Carlo (MC) sampling. In Section 3.2, general characteristics of MC sampling are described. In Section 3.3, the characteristics of Robustness optimisation based on MC sampling are described. It is shown that for each element of the domain, the MC estimate function is either discontinuous or constant. These characteristics are unfavourable for optimisation. Section 3.3 illustrates the observations that triggered the research for alternative Robustness estimation methods with better optimisation characteristics. The following characteristics are considered relevant for Robustness estimation methods in the context of Robustness optimisation.

- Effectiveness. An estimation method is considered effective if the estimation result is unbiased.
- Efficiency. The proposed measure for efficiency of an estimate, is the standard error ($se$) of the Robustness estimator. The goal is to estimate Robustness at an accuracy level of at least some *t*hres*h*old value $se^{[th]}$. The idea is that, in expectation, the Robustness estimator with the lowest standard error will need the smallest number of samples to satisfy the accuracy level of at most $se^{[th]}$ and is therefore most efficient.
- Continuity of the Robustness estimate function.
- Derivative information generated by the estimate function.

The first alternative method is called the Smoothed Monte Carlo (SMC) method and is discussed in Section 3.4. The general idea of the SMC method is to use sample information in such a way that it results in a continuous Robustness estimate function.

The second alternative method is called the N-1 Monte Carlo (N-1MC) method and is discussed in Section 3.5. The general idea of the N-1MC method is to sample N-1 of the N elements of $\boldsymbol{v}$ and use the cumulative distribution function of the not sampled element to compute the Robustness estimate. It is shown that this method gives an unbiased Robustness estimate with better optimisation characteristics than the MC method.

The third alternative method is called the Directional Sampling method and is discussed in Section 3.6. The DS method is applicable in the situation that $\boldsymbol{v}$ follows a so-called spherical symmetric distribution, like the multivariate Normal distribution and gives unbiased Robustness estimates with better optimisation characteristics than the MC method.

The fourth alternative method is called the Exponential Simplex (ES) method and is discussed in Section 3.7. The ES method is applicable in the situation that $\boldsymbol{v}$ follows a multivariate Exponential distribution and gives unbiased Robustness estimates with better optimisation characteristics than the MC method.

Finally, so-called Set Bounding (SB) methods are discussed in Section 3.8. The general idea of the SB methods is to inscribe or circumscribe the Happy set with a set for which the probability mass can be relatively easily computed. This way, the SB methods result in lower bounds and upper bounds for the Robustness.

First, the Monte Carlo method is introduced. It is the reference method in this thesis to which the alternative Robustness estimation methods are compared.

## 3.2 Monte Carlo estimation method

During World War II, John von Neumann and Stanislaw Ulam developed a simulation method for studying the extent to which neutrons can travel through various materials. Since their studies were classified, von Neumann gave it the code Monte Carlo method (see Ghahramani, 2000). The MC method can be described as follows. Consider the Happy set indicator function $I$ as defined in (2.5) and i.i.d. random vectors $\boldsymbol{v}^{[m]}$ and $m = 1, .., \mathrm{M}$ with some given probability distribution. The Monte Carlo Robustness estimator is defined as

$$\overset{\text{mc}}{\boldsymbol{R}}(x) = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} I(\boldsymbol{v}^{[m]}, x) \tag{3.1}$$

The standard error of the Monte Carlo Robustness estimator is

$$se\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right) = \left[var\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)\right]^{\frac{1}{2}} = \sqrt{\frac{1}{\mathrm{M}}\left(R(x) - R(x)^2\right)} \tag{3.2}$$

and is based on the definitions in (Rice, 1995). The following generalisation is used.

**Definition 3.1** *The Monte Carlo Robustness estimate function* $\overset{mc}{R} : \mathbb{X} \longrightarrow \left\{0, \frac{1}{M}, \frac{2}{M}, .., 1\right\}$, *given a realisation* $v^{[1]}, .., v^{[M]}$ *of random vectors* $\boldsymbol{v}^{[1]}, .., \boldsymbol{v}^{[M]}$ *is defined as*

$$\overset{mc}{R}(x) = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} I(v^{[m]}, x) \tag{3.3}$$

**Corollary 3.1** *The estimated standard error* $\hat{se}\left(\overset{mc}{\boldsymbol{R}}(x)\right)$ *of the Robustness estimator is*

$$\hat{se}\left(\overset{mc}{\boldsymbol{R}}(x)\right) = \sqrt{\frac{1}{\text{M-1}}\left(\overset{mc}{R}(x) - \overset{mc}{R}(x)^2\right)} \tag{3.4}$$

**Corollary 3.2** *If* $\exists x^{[1]}, x^{[2]}$ *such that* $\overset{mc}{R}(x^{[1]}) \neq \overset{mc}{R}(x^{[2]})$ *then the MC estimate function* $\overset{mc}{R}$ *is a discontinuous function.*

Example 3.1 introduces a Robustness Programming problem and illustrates the stepwise constant character of the MC estimate function.

**Example 3.1** *Consider random vector* $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix}$ *and Robustness function* $R : \mathbb{R}^2 \longrightarrow [0,1]$ *with*

$$R(x) = \Pr\left\{\mathrm{L} \leq \begin{bmatrix} x_1 & 0.8 - x_2 \\ 0.7 - 2x_1 & x_2 \end{bmatrix} (T\boldsymbol{v} + \mu) \leq \mathrm{H}\right\} \tag{3.5}$$

*where*

$$\mathrm{L} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mathrm{H} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \mu = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

*Two situations are studied: In the so-called "Gaussian" situation the elements of* $\boldsymbol{v}$ *are i.i.d. and follow a Gaussian*[1] *distribution; In the so-called "Exponential" situation the elements of* $\boldsymbol{v}$ *are i.i.d. and follow the Exponential distribution. Figure 3.1 shows the MC estimate of* (3.5) *as a function of* $x$. *The mesh surface on the left-hand side is based on i.i.d samples of* $\boldsymbol{v}_n^{[m]} \sim Exp(1)$ *with* $n = 1,2$ *and* $m = 1,..,20$. *The mesh surface on the right-hand side is based on i.i.d samples of* $\boldsymbol{v}_n^{[m]} \sim N(0,1)$ *with* $n = 1,2$ *and* $m = 1,..,20$. *Consider the mesh surface in Figure 3.1 of the Gaussian situation (i.e. the right-hand side). Around the point* $x = (0.5, -0.3)$, *the mesh surface appears to be flat. Figure 3.2 gives a graphical representation of the discontinuity of the MC estimate function, where* $x_2 = -0.3$ *and the line pieces between open (*$\circ$*) and closed (*$\bullet$*) edges depict* $\overset{\text{mc}}{R}(x_1, -0.3)$.

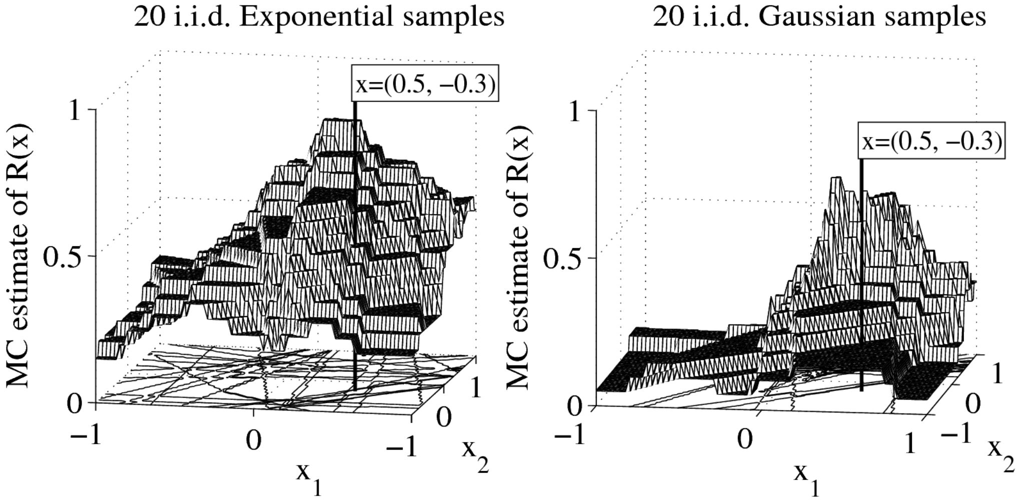


Figure 3.1: Illustration of MC estimation in Exponential and Gaussian situation; 20 samples

To facilitate the discussion, the discontinuity of $\overset{\text{mc}}{R}$ is elaborated by introducing the concept of Upper Semi Continuity[2] (USC). Among others, Folland (1999) shows that an indicator function of a closed set is Upper Semi Continuous. In 2.2 the Happy set is defined as a closed set.

[1] The Normal distribution is also called the Gaussian distribution (Weisstein, 1998).

[2] A function $f : \mathbb{X} \longrightarrow \mathbb{R}$ is USC in point $a \in \mathbb{X}$ if: $\forall \varepsilon > 0\ \exists \delta > 0$ such that $y \in \mathbb{X}$ and $\|a - y\| \leq \delta$ imply that $f(a) - f(y) > -\varepsilon$ (Bazaraa et al., 1993).

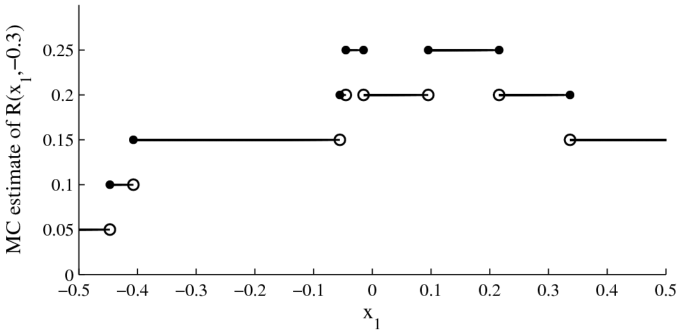


Figure 3.2: Illustration of MC estimation in Gaussian situation; $x_2 = -0.3$; 20 samples

**Corollary 3.3** *The Happy set indicator function $I(v, x)$ is an USC function of $x$ for given $v$.*

Folland (1999) shows that the sum of USC functions is USC. The estimate $\overset{\text{mc}}{R}(x)$ is a sum of USC (Happy set indicator) functions. This gives Corollary 3.4.

**Corollary 3.4** *The Monte Carlo Robustness estimate function $\overset{mc}{R}(x)$ is an USC function.*

## 3.3 Robustness optimisation using MC estimates

The Robustness $R(x)$ can be estimated by $\overset{\text{mc}}{R}(x)$ and likewise an estimate for the optimal Robustness is

$$\overset{\text{mc}}{R}{}^{*} = \max_{x \in \mathbb{X}} \overset{\text{mc}}{R}(x) \tag{3.6}$$

An approach for the optimisation of $\overset{\text{mc}}{R}(x)$, is the use of iterative algorithms such as SQP (as discussed in Venkataraman, 2002; Schittkowski, 2005) that generate trial points and compute their function values $\overset{\text{mc}}{R}(x^{[1]})$, $\overset{\text{mc}}{R}(x^{[2]})$,$\overset{\text{mc}}{R}(x^{[3]})$,...,etc. to approximate the maximum.

The approach in this thesis is to generate sample values $v^{[1]}, .., v^{[M]}$ before optimisation and compute each trial point during the iterative optimisation using the same sample values. This method is known as the *external sampling method* (Mak et al., 1999). Similar ideas are introduced by Kleywegt et al. (2001) and by Gurkan et al. (1999), which are respectively called *sample average approximation* and *sample-path optimisation*.

An iterative optimisation method that is based on (approximated) gradient and Hessian information can fail to converge to the (local) maximum due to discontinuities in the objective function or restriction function(s) (Kelley, 1999). This bad convergence occurs due to the piece-wise constant character of the MC estimate function.

Let $\mathbb{C} \subseteq \mathbb{X}$ be the subset of the domain for which $\overset{\text{mc}}{R}(x)$ is continuous. The finite range of the MC estimate function implies that the gradient is $\nabla \overset{\text{mc}}{R}(x) = 0$ for $x \in \mathbb{C}$. All $x \in \mathbb{C}$

correspond to local maxima of $\overset{\text{mc}}{R}(x)$. An iterative solver can get *trapped* in each of these local maxima and stop before reaching the global maximum.

From Figure 3.1 it can be concluded that, relative to the point $x = (0.5, -0.3)$, lower values of $x_1$ and higher values of $x_2$ will lead to higher values of $\overset{\text{mc}}{R}(x)$. An optimisation algorithm does not have global information. An optimisation algorithm performs locally a finite number of function evaluations to find improving directions for the objective function. The gradient of the objective function can be used to determine a search direction leading to higher objective function values. An important concept in this context is known as Finite Differencing (FD) (Kelley, 1999). Finite Differencing is a method to estimate the gradient of a function using a finite number of function evaluations. A finite difference gradient estimate of function $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ with FD step-size $\Delta$ is $\hat{\nabla} f(x) = \frac{1}{\Delta}\begin{bmatrix} f(x_1+\Delta, x_2) - f(x_1, x_2) \\ f(x_1, x_2+\Delta) - f(x_1, x_2) \end{bmatrix}$. For larger $\Delta$, the points $(x_1, x_2)$,$(x_1 + \Delta, x_2)$ and $(x_1, x_2 + \Delta)$ are farther apart and result in a less accurate gradient estimate. However, in the situation of a step function like $\overset{\text{mc}}{R}$, we do not want to be precise. Instead we want $\Delta$ to be large enough to *detect* in which direction $\overset{\text{mc}}{R}(x)$ increases. Example 3.2 illustrates that the use of step-sizes in an optimisation algorithm, is a sensitive factor when optimising $\overset{\text{mc}}{R}(x)$.

**Example 3.2** *If we take* $\Delta = 0.1$ *in the Gaussian situation, then* $\overset{mc}{R}(0.5, -0.3) = \overset{mc}{R}(0.5+\Delta, -0.3) = \overset{mc}{R}(0.5, -0.3+\Delta) = 0.15$ *from which follows that* $\hat{\nabla}\overset{mc}{R}(0.6, -0.6) = [0, 0]^{\mathsf{T}}$. *Such gradient value satisfies the first order optimality condition and causes the optimisation algorithm to stop at this local optimum. Alternatively, consider a less precise FD estimate based on* $\Delta = 0.2$. *Since* $\overset{mc}{R}(0.5 + \Delta, -0.3) = 0.1$ *and* $\overset{mc}{R}(0.5, -0.3 + \Delta) = 0.2$ *the FD gradient estimate is* $\hat{\nabla}\overset{mc}{R}(0.5, -0.3) = [-0.25, 0.25]^{\mathsf{T}}$. *From Figure 3.1 can be concluded that this gradient direction corresponds to higher values of* $\overset{mc}{R}(x)$.

Example 3.3 illustrates optimisation of the MC estimate function, in relation to the sample size and FD step-size.

**Example 3.3** *An approach to make the MC estimate function better suitable for optimisation, is to increase the sample size. Figure 3.3 illustrates that the mesh surfaces based on 1000 samples are less* rough *than the mesh surfaces in Figure 3.1 based on 20 samples. Although the* $\overset{mc}{R}(x)$ *function is discontinuous for any integer M, the mesh surfaces based on 1000 samples almost* look *like surfaces of a continuous functions. The influence of the FD step-size and number of samples on Robustness optimisation, is illustrated in this example as follows: The RP problem*

$$\overset{mc}{R}^{*} = \max_{-1 \leq x_1, x_2 \leq 1} \overset{mc}{R}(x) \tag{3.7}$$

*is solved with the so-called* FMINCON[3] *solver of the Matlab Optimisation Toolbox®, which is based on an SQP algorithm (Venkataraman, 2002). The corresponding optimal design*

[3] The FMINCON solver is typically used for constrained NLP problems. Since the Robustness optimisation cases discussed in Chapter 5 are constrained NLP problems, the FMINCON solver is an obvious choice. The FMINUNC solver of the Matlab Optimisation Toolbox is an NLP solver for unconstrained

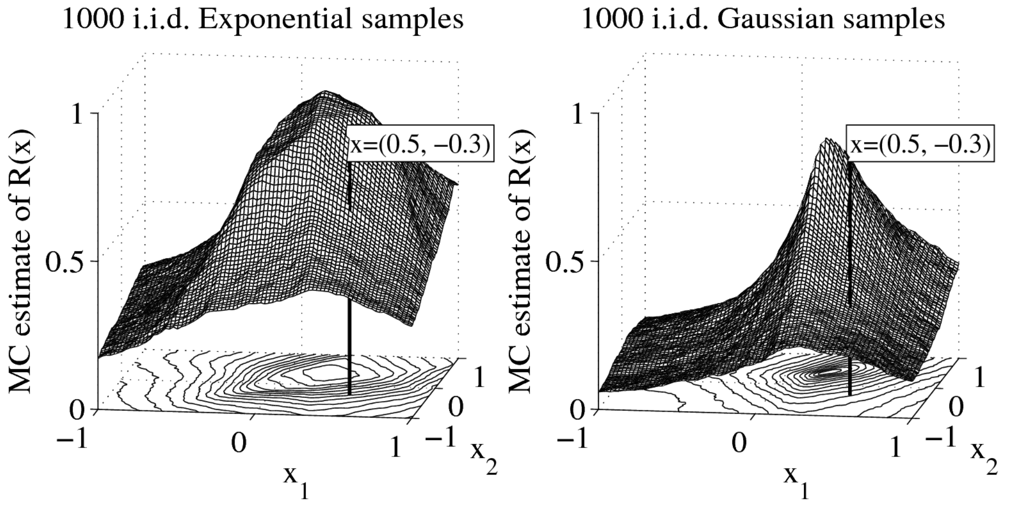


Figure 3.3: Illustration of MC estimation in Exponential and Gaussian situation; 1000 samples

*is* $x^* = \underset{-1\leq x_1,x_2\leq 1}{\arg\max} \overset{mc}{R}(x)$. *All parameters are kept at the Matlab default values. The starting point is* $x^{[0]} = (0.5, -0.3)$. *The Robustness optimisation is based on the following values of the FD step-size*[4]*:* $\Delta \in \left\{ \left(\frac{1}{2}\right)^{10}, \left(\frac{1}{2}\right)^{9}, ..., \left(\frac{1}{2}\right)^{1}, \left(\frac{1}{2}\right)^{0} \right\}$. *The* $\overset{mc}{R}{}^*$ *column in Table 3.1 presents the values to which the SQP algorithm converges, for a given* $\Delta$. *A highly accurate estimate* $\tilde{R}(x^*)$ *is computed, to get an idea of the actual Robustness* $R(x^*)$ *and to compare results. The highly accurate estimate* $\tilde{R}(x^*)$ *is defined in Appendix A.3. As mentioned in the beginning of this section, the SQP algorithm can converge to local optima. Therefore, the values in the* $\overset{mc}{R}{}^*$ *column of Table 3.1 are the best values found by the SQP solver, but are not necessarily global optima of* (3.7).

*The best solutions in Table 3.1 are given in bold. The corresponding results are illustrated in Figures 3.4, 3.5, 3.6 and 3.7. Each figure illustrates the optimisation trajectory, the Happy set at the starting point* $x^{[0]}$ *and the Happy set given the design* $x^*$ *to which the SQP algorithm converges.*

From these results we can conclude that increasing the number of samples has a positive effect on making the MC estimate function more suitable for optimisation. However, increasing the number of samples comes at a price: The computation time goes up proportional to the number of samples. The results in Table 3.3 show that for most of the step-sizes, the found optima are unsatisfactory. Furthermore, these results show that changing the number of samples, also changes the optimal setting for the FD step-size. It

problems, based on a BFGS algorithm (Venkataraman, 2002). All experiments of this chapter (3) have been verified with the BFGS algorithm, where the constraint $-1 \leq x_1, x_2 \leq 1$ was not taken into account. Only the SQP results are presented in this chapter, because all results based on the BFGS algorithm turned out to be close to the results of the SQP algorithm.

[4] Step-size values $\Delta > 1$ caused de SQP algorithm to fail; Such step-size is too big relative to the design space $-1 \leq x_1, x_2 \leq 1$. Step-size values $\Delta < \left(\frac{1}{2}\right)^{10}$ appeared too small and caused the SQP algorithm to stop at the first iteration and return $\overset{mc}{R}(x^{[0]})$ as the best solution.

Table 3.1: SQP Optimisation results of MC estimate function, for a given $\Delta$, sample size and distribution type

| | Exponential case | | | | Gaussian case | | | |
|---|---|---|---|---|---|---|---|---|
| | M=20 | | M=1000 | | M=20 | | M=1000 | |
| $\Delta$ | $\overset{\text{mc}}{R}^*$ | $\tilde{R}\,(x^*)$ | $\overset{\text{mc}}{R}^*$ | $\tilde{R}\,(x^*)$ | $\overset{\text{mc}}{R}^*$ | $\tilde{R}\,(x^*)$ | $\overset{\text{mc}}{R}^*$ | $\tilde{R}\,(x^*)$ |
| Robustness estimate at starting point $x^{[0]} = (0.5, -0.3)$: | | | | | | | | |
| | 0.500 | 0.642 | 0.614 | 0.642 | 0.150 | 0.293 | 0.294 | 0.293 |
| $\left(\frac{1}{2}\right)^{10}$ | 0.500 | 0.642 | 0.818 | 0.831 | 0.150 | 0.293 | 0.294 | 0.293 |
| $\left(\frac{1}{2}\right)^{9}$ | 0.500 | 0.642 | 0.853 | 0.876 | 0.150 | 0.293 | 0.294 | 0.293 |
| $\left(\frac{1}{2}\right)^{8}$ | 0.500 | 0.642 | 0.856 | 0.867 | 0.150 | 0.293 | 0.464 | 0.451 |
| $\left(\frac{1}{2}\right)^{7}$ | 0.500 | 0.642 | **0.954** | **0.959** | 0.150 | 0.293 | 0.580 | 0.574 |
| $\left(\frac{1}{2}\right)^{6}$ | 0.500 | 0.642 | 0.954 | 0.959 | 0.150 | 0.293 | **0.788** | **0.769** |
| $\left(\frac{1}{2}\right)^{5}$ | 0.500 | 0.642 | 0.953 | 0.958 | 0.150 | 0.293 | 0.780 | 0.759 |
| $\left(\frac{1}{2}\right)^{4}$ | 0.500 | 0.642 | 0.946 | 0.952 | 0.150 | 0.293 | 0.766 | 0.742 |
| $\left(\frac{1}{2}\right)^{3}$ | 0.650 | 0.811 | 0.953 | 0.959 | 0.150 | 0.293 | 0.779 | 0.759 |
| $\left(\frac{1}{2}\right)^{2}$ | **0.800** | **0.926** | 0.912 | 0.923 | **0.400** | **0.561** | 0.714 | 0.696 |
| $\left(\frac{1}{2}\right)^{1}$ | 0.750 | 0.896 | 0.835 | 0.849 | 0.300 | 0.482 | 0.439 | 0.445 |
| $\left(\frac{1}{2}\right)^{0}$ | 0.500 | 0.669 | 0.681 | 0.671 | 0.250 | 0.329 | 0.321 | 0.329 |
| Average computation time and average estimated standard error per function evaluation are: | | | | | | | | |
| time (sec.) | 0.0033 | 129.223 | 0.1464 | 55.947 | 0.0033 | 138.505 | 0.1455 | 158.013 |
| $\hat{se}$ | 0.1068 | 0.0005 | 0.0143 | 0.0005 | 0.0744 | 0.0005 | 0.0129 | 0.0005 |

Comments on Table 3.1:

1. $\overset{\text{mc}}{R}^*$ is the Robustness estimate based on M samples, of point $x^*$ where SQP converged to.
2. $\tilde{R}(x^*)$ is an MC estimate of $R(x^*)$, with $\hat{se}\left(\tilde{\boldsymbol{R}}\,(x^*)\right) < 0.0005$ (See Appendix A.3).

appears that satisfactory optimisation of the MC estimate function depends on the FD step-size value.

The Figures 3.4, 3.5, 3.6 and 3.7 show that the optimisation trajectories change if the number of samples changes and ultimately converge to different solutions. Increasing the sample size M, from 20 to 1000, decreases the estimated standard error on average. The results support the following assumption: Optimisation based on a more accurate Robustness estimate function (i.e. with a lower standard error) leads to better results $\tilde{R}(x^*)$. In this regard, two aspects can be distinguished

1. From Definition (3.3) it follows that $\overset{\text{mc}}{R}(x)$ based on M=20 samples is a different function than $\overset{\text{mc}}{R}(x)$ based on M=1000 samples. Therefore, $\overset{\text{mc}}{R}^*$ and $x^*$ depend on the value of M. From the weak law of large numbers (Grimmett and Stirzaker, 2001) follows that $\overset{\text{mc}}{R}(x) \longrightarrow R(x)$ for M$\longrightarrow \infty$, for a given $x$. Consequently, $\overset{\text{mc}}{R}^* \longrightarrow R^*$ for M$\longrightarrow \infty$ .

2. The Figures 3.4 and 3.6 show that the SQP optimisation algorithm fails to converge to $\overset{\text{mc}}{R}^*$ if M=20. The Figures 3.5 and 3.7 suggest a better convergence to $\overset{\text{mc}}{R}^*$ if M=1000.

The observations in this section trigger the search for alternative Robustness estima-

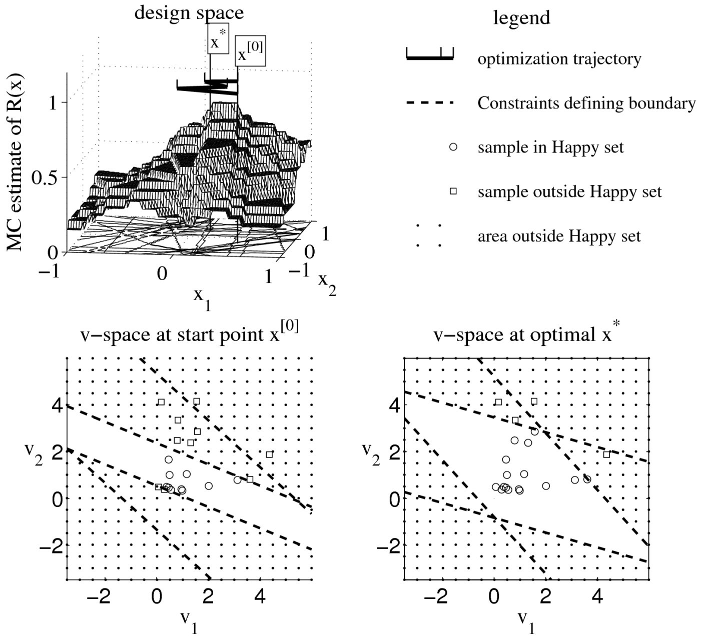


Figure 3.4: SQP MC estimate function optimisation results; Exponential situation; M=20; $\Delta = \left(\frac{1}{2}\right)^2$; The unmarked white area between the dashed line in the v-space depicts the Happy set

tion methods, that are continuous, more smooth and more accurate than the MC estimate function. Focus is on two types of methods: In Sections 3.4, 3.5, 3.6 and 3.7 methods are discussed that estimate $R(x)$ via sampling techniques and in Section 3.8 methods are discussed for computing lower and upper bounds of $R(x)$.

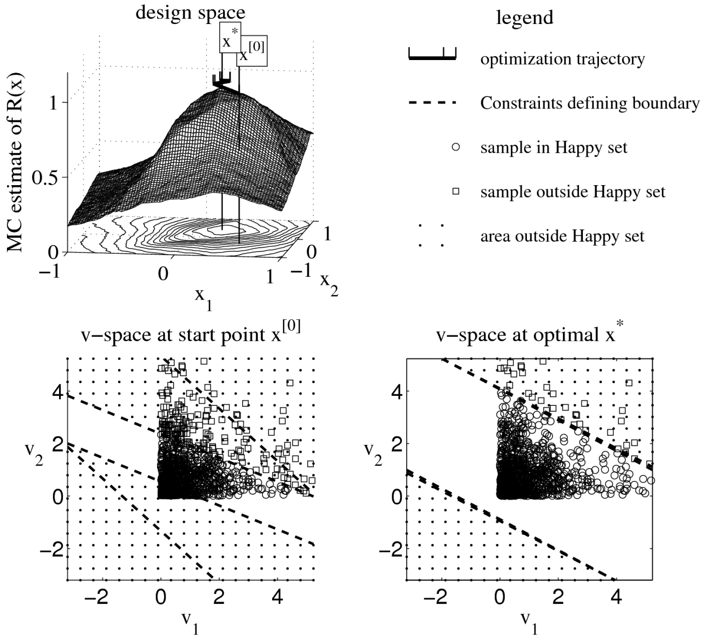


Figure 3.5: SQP MC estimate function optimisation results; Exponential situation; M=1000;$\Delta = \left(\frac{1}{2}\right)^7$; The unmarked white area between the dashed line in the v-space depicts the Happy set

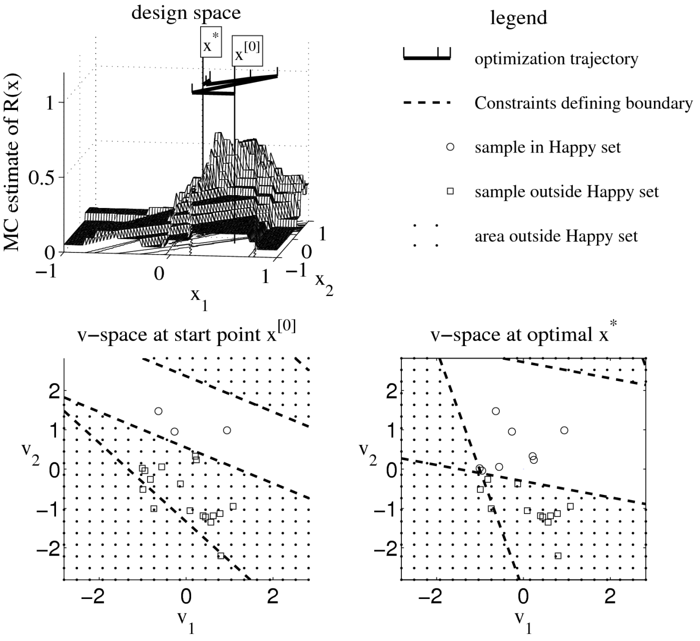


Figure 3.6: SQP MC estimate function optimisation results; Gaussian situation; M=20;$\Delta = \left(\frac{1}{2}\right)^2$; The unmarked white area between the dashed line in the v-space depicts the Happy set

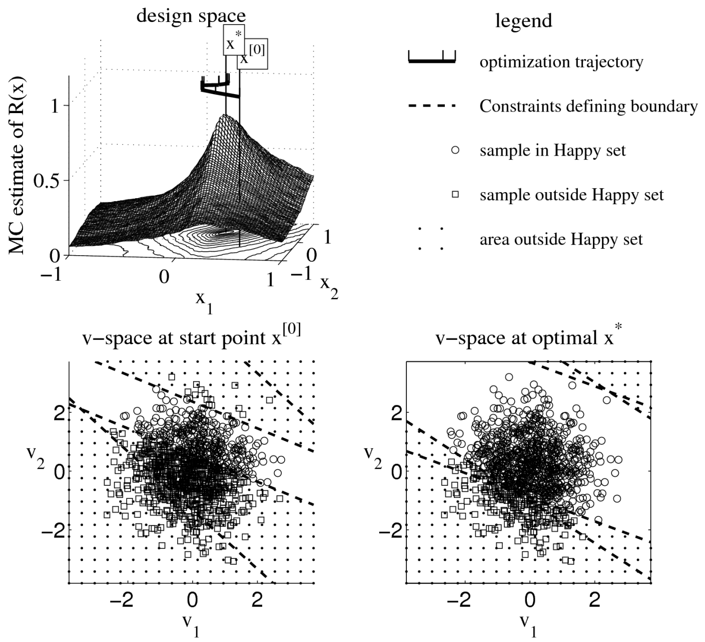


Figure 3.7: SQP MC estimate function optimisation results; Gaussian situation; M=1000;$\Delta = \left(\frac{1}{2}\right)^6$; The unmarked white area between the dashed line in the v-space depicts the Happy set

## 3.4 Smoothed Monte Carlo estimation method

The discontinuity of the MC estimate function is inconvenient for optimisation. In this section, a smoothing method is introduced that neutralises discontinuities.

The idea behind Smoothed Monte Carlo (SMC) estimation is to add a "smoothing" function $s(x)$ to the Monte Carlo estimate that produces a continuous function $\overset{\text{smc}}{R}(x)$. These concepts are discussed in Hendrix and Olieman (2008). The set $\mathbb{C} \subseteq \mathbb{X}$ is defined as the subset of the domain for which $\overset{\text{mc}}{R}(x)$ is continuous. Focus is on a smoothing function such that $\overset{\text{smc}}{R}$ is continuous on the sub-domain $\mathbb{S}$ with $\mathbb{C} \subset \mathbb{S} \subset \mathbb{X}$ that is sufficient to make the method work in practice. The SMC estimate function is not unbiased and deviates from the unbiased MC estimate function. However, it will be shown that $\forall x \in \mathbb{S}$, $\left|\overset{\text{mc}}{R}(x) - \overset{\text{smc}}{R}(x)\right| \leq \frac{1}{2\text{M}}$ , which can be set arbitrary close to zero, by choosing $M$ (the number of samples) sufficiently large.

In the following, the set $\mathbb{S}$ and the smoothing function $s(x)$ are defined for which $\overset{\text{smc}}{R} : \mathbb{S} \longrightarrow [0,1]$ with $\overset{\text{smc}}{R}(x) = \overset{\text{mc}}{R}(x) + s(x)$ is continuous at each $x \in \mathbb{S}$. The definition of the set $\mathbb{S}$ is based on three concepts, respectively called *cliff-points*, *n-Happy-Boundary point* and *standard MC discontinuity point*:

Let $\overset{\text{mc}}{R}$ be discontinuous in point $a$. There exists a point $x \neq a$ arbitrarily close to $a$, such that $|\overset{\text{mc}}{R}(x) - \overset{\text{mc}}{R}(a)|$ is a multiple of $\frac{1}{\text{M}}$. Graphically speaking, there is a steep function change around point $a$. Such steep function change can be seen as a *cliff* and such discontinuity points are called $\alpha-$*cliff points*, where $\alpha \in \left\{\frac{1}{\text{M}}, \frac{2}{\text{M}}, .., 1\right\}$ corresponds to the *magnitude* of the jump:

**Definition 3.2** *Let $f : \mathbb{R}^{\mathrm{I}} \longrightarrow \mathbb{R}$, $\alpha \geq 0$. Point $a \in \mathbb{R}^{\mathrm{I}}$ is an $\alpha$-cliff point of $f$, if $\forall\delta > 0$ $\exists y : \|a - y\| \leq \delta$ and $|f(a) - f(y)| = \alpha$.*

**Corollary 3.5** *Let $f$ be an USC function and $a \in \mathbb{R}^{\mathrm{I}}$ be an $\alpha$-cliff point of $f$, then $\forall\delta > 0$ $\exists y : \|a - y\| \leq \delta$ and $f(a) - f(y) = \alpha$.*

**Proof.** Let $\varepsilon = \frac{1}{2}\alpha$. The USC property of $f$ says $\exists\delta > 0, \forall y : \|a - y\| \leq \delta \implies f(a) - f(y) > -\varepsilon = -\frac{1}{2}\alpha$. This USC property excludes the possibility that for $\alpha$-cliff point $a$, $\forall\delta > 0$ $\exists y : \|a - y\| \leq \delta$ and $f(a) - f(y) = -\alpha$. ■

**Corollary 3.6** *Let $a \in \mathbb{X}$ be an $\alpha$-cliff point of $\overset{mc}{R}$, with $\alpha \in \left\{\frac{1}{M}, \frac{2}{M}, .., 1\right\}$. $\overset{mc}{R}$ is discontinuous in $a$.*

Let $v^{[1]}, ..., v^{[\mathrm{M}]}$ be M independent realisations of $\boldsymbol{v}$. The step-function $\overset{\mathrm{mc}}{R}(x)$ makes a jump, as $x$ passes a discontinuity point $a \in \mathbb{X} \setminus \mathbb{C}$. The magnitude of this jump corresponds to $\frac{n}{\mathrm{M}} = \alpha$ where $n$ is the number of samples entering or leaving the Happy set simultaneously. The Happy set is defined as a closed set. Hence, samples on the boundary of the Happy set are part of the Happy set. The Happy set is defined by functions $u_s(x, v)$ which are continuous in $x \in \mathbb{R}^{\mathrm{I}}$ for a given $v \in \mathbb{V}$. Consequently, there are at least $n$ samples of $\boldsymbol{v}$ on the boundary[5] $\partial\mathbb{H}(a)$ at the discontinuity point $a$. The number of samples on the boundary of the Happy set determines the maximum possible magnitude of the jump of the step function and characterises the type of discontinuity.

**Definition 3.3** *An $n$-Happy-Boundary point is a point $a \in \mathbb{X}$ for which the boundary $\partial\mathbb{H}(a)$ of the Happy set contains $n$ out of* M *samples of $\left\{v^{[1]}, .., v^{[\mathrm{M}]}\right\}$, i.e.*

$$\left|\left\{v^{[1]}, .., v^{[\mathrm{M}]}\right\} \cap \partial\mathbb{H}(a)\right| = n$$

Note that a 1-Happy-Boundary is not necessarily a $\frac{1}{\mathrm{M}}$-cliff point. This is the case, if there is a sample on the boundary $\partial\mathbb{H}(a)$ that does not leave the Happy set in a small environment of $a$, i.e. $\exists\delta > 0, m$ with $v^{[m]} \in \partial\mathbb{H}(a)$ and $\forall x \in \mathbb{B}(a, \delta)$ $v^{[m]} \in \mathbb{H}(x)$ where $\mathbb{B}(a, \delta) = \left\{x \in \mathbb{R}^{\mathrm{I}} \middle| \|a - x\| \leq \delta\right\}$ is a ball around $a$ with radius $\delta$. Moreover, a $\frac{1}{\mathrm{M}}$-cliff point is not necessarily a 1-Happy-Boundary point. For example, if there are two samples on the boundary $\partial\mathbb{H}(a)$ and one of these two samples is in the Happy set in a small environment of $a$.

**Definition 3.4** *A standard MC discontinuity point, is a point $a \in \mathbb{X}$ that is both a $\frac{1}{\mathrm{M}}$-cliff point as well as a 1-Happy-Boundary point*

Let $a \in \mathbb{X}$ be a standard MC discontinuity point. Let $v^{[m^*]}$ be the sample on the boundary $\partial\mathbb{H}(a)$. Consequently, all other samples $v^{[m]}$ with $m \neq m^*$ are either in the interior of the Happy set or outside the Happy set. The step-function $\overset{\mathrm{mc}}{R}$ *jumps* with a magnitude of $\frac{1}{\mathrm{M}}$ in an arbitrary small interval around point $a$. This $\frac{1}{\mathrm{M}}$ function value change, corresponds to $v^{[m^*]}$ leaving the Happy set, while all the other samples $v^{[m]}$, $m \neq m^*$ do not touch or cross the boundary of the Happy set.

[5] Consider a set $\mathbb{H}$. The notation of Bazaraa et al. (1993) for respectively the boundary, the interior and the closure are $\partial\mathbb{H}$, $\mathrm{int}\mathbb{H}$ and $\mathrm{cl}\mathbb{H}$.

The word "standard" in the name "standard MC discontinuity point" is used, because during the experiments illustrated in Chapter 5, this type of points were encountered. Other points where $\overset{\text{mc}}{R}$ is discontinuous do exist, but these are considered to be of less practical relevance. In Appendix A.4 the Advanced Smoothed Monte Carlo (ASMC) algorithm is introduced, which generalises the SMC method, such that the estimation function is continuous at each evaluated trial point in $\mathbb{X}$. However, the ASMC algorithm is not further elaborated, since the SMC method works sufficiently well for the studied cases. For the sake of completeness (and possible future use) we pay attention to the ASMC algorithm in Appendix A.4.

**Definition 3.5** *The domain* $\mathbb{S}$ *of* $\overset{smc}{R}$ *is defined as*

$$\mathbb{S} = \left\{ x \in \mathbb{R}^I \middle| \tfrac{1}{\mathrm{M}} \le \overset{mc}{R}(x) \le \tfrac{\mathrm{M}-1}{\mathrm{M}} \text{ and } \left|\left\{v^{[1]}, .., v^{[\mathrm{M}]}\right\} \cap \partial\mathbb{H}(x)\right| \le 1 \right\} \tag{3.8}$$

From the definition it follows that for each $x \in \mathbb{S}$ there is at least one sample in the Happy set and at least one sample not in the Happy set and at most one sample is on the boundary of the Happy set. The practical relevance of the set $\mathbb{S}$ is discussed at the end of this section. In the following, the smoothing function is introduced and continuity of $\overset{\text{smc}}{R}(x)$ for $x \in \mathbb{S}$ is shown.

The smoothing function $s(x)$ is constructed by means of three functions, namely $g : \mathbb{S} \times \mathbb{V} \longrightarrow \mathbb{R}$, $d^{[in]} : \mathbb{S} \longrightarrow \mathbb{R}_+$ and $d^{[out]} : \mathbb{S} \longrightarrow \mathbb{R}_{++}$. We first introduce.

$$g(x, v) = \min_s \left\{ \min \left\{ \mathrm{H}_s - u_s(x, v), u_s(x, v) - \mathrm{L}_s \right\} \right\}$$

which represents the *slack* of the uncertain restriction that is violated the most ($g(x, v) < 0$), or is closest to violation ($g(x, v) \ge 0$). Function $g(x, v)$ has the following properties:

$$v \in \partial\mathbb{H}(x) \Longleftrightarrow g(x, v) = 0 \tag{3.9}$$

$$v \in \mathbb{H}(x) \Longleftrightarrow g(x, v) \ge 0 \tag{3.10}$$

$$v \notin \mathbb{H}(x) \Longleftrightarrow g(x, v) < 0 \tag{3.11}$$

$$b \in \mathbb{X} \text{ is a } n\text{-Happy-Boundary point} \Longleftrightarrow \left|\left\{ m | g(b, v^{[m]}) = 0 \right\}\right| = n \tag{3.12}$$

$$\overset{\text{mc}}{R}(x) = \frac{1}{\mathrm{M}} \left| \left\{ m | g(x, v^{[m]}) \ge 0 \right\} \right| \tag{3.13}$$

Note that $g(x, v)$ is continuous in $x$, for given $v$ since $u_s(x, v)$ is continuous in $x$ as assumed in Section 2.1.

**Corollary 3.7** *Each* $\frac{1}{\mathrm{M}}$*-cliff point* $a \in \mathbb{S}$ *of* $\overset{mc}{R}$ *is an n-Happy-Boundary point with* $n \ge 1$.

**Proof.** Proof by contradiction: Let $a \in \mathbb{S}$ be an $\frac{1}{\mathrm{M}}$-cliff point and also a 0-Happy-Boundary point. As $a$ is a 0-Happy-Boundary point, (3.13) holds in a *strict* sense:

$$\mathrm{M}\,\overset{\text{mc}}{R}(a) = \left| \left\{ m | g(a, v^{[m]}) > 0 \right\} \right|$$

For a continuous function $g$, $\exists \epsilon > 0$ such that $\forall y : \|a - y\| \le \epsilon$ holds that $\left| \left\{ m | g(y, v^{[m]}) > 0 \right\} \right| = \mathrm{M}\,\overset{\text{mc}}{R}(a)$, which contradicts $a$ being an $\frac{1}{\mathrm{M}}$-cliff point. ■

**Corollary 3.8** *An n-Happy-Boundary point is an $\alpha$-cliff point with $\alpha \leq \frac{n}{\mathrm{M}}$*

**Corollary 3.9** *Let $\overset{mc}{R}$ be discontinuous at $x \in \mathbb{S}$. Point $x$ is a standard MC discontinuity point.*

Throughout the remainder of this section, $a \in \mathbb{S}$ is a standard MC discontinuity point. Properties (3.9),(3.10) and (3.11) suggest that if $g(x, v)$ is close to zero, then the Euclidean distance between sample $v$ and $\partial\mathbb{H}(x)$ is also close to zero[6]. The following functions are related to the distance of the samples closest to the boundary, respectively in the Happy set and outside the Happy set. Let

$$d^{[in]}(x) \quad = \quad \min_{m} \quad \left\{ g\left(x, v^{[m]}\right) \middle| \, g\left(x, v^{[m]}\right) \geq 0 \right\} \tag{3.14}$$

correspond to the sample *in* the Happy set with the least non-negative slack. Let

$$d^{[out]}(x) \quad = \quad \min_{m} \quad \left\{ -g\left(x, v^{[m]}\right) \middle| \, g\left(x, v^{[m]}\right) < 0 \right\} \tag{3.15}$$

correspond to the sample *not in* the Happy set with the least negative slack. Note the following properties

$$d^{[in]}(a) = 0 \tag{3.16}$$

$$d^{[in]}(x) \geq 0 \text{ for all } x \in \mathbb{S} \tag{3.17}$$

$$d^{[out]}(x) > 0 \text{ for all } x \in \mathbb{S} \tag{3.18}$$

$$d^{[in]} \text{ and } d^{[out]} \text{ are discontinuous in point } a \tag{3.19}$$

The functions $d^{[in]}$ and $d^{[out]}$ lead to the following definition of the smoothing function:

**Definition 3.6** *The smoothing function $s : \mathbb{S} \longrightarrow [-\frac{1}{2\mathrm{M}}, \frac{1}{2\mathrm{M}})$ is defined as:*

$$s(x) = \frac{1}{2\mathrm{M}} \left( \frac{2d^{[in]}(x)}{d^{[in]}(x) + d^{[out]}(x)} - 1 \right) \tag{3.20}$$

The idea behind the smoothing function $s$ is the following. It can be shown that $a$ is a $\frac{1}{\mathrm{M}}$-cliff point of function $s$ with a $\frac{1}{\mathrm{M}}$ step in the opposite direction of $\overset{mc}{R}$ such that it neutralises the discontinuity. The neutralisation of the discontinuity is illustrated in Figure 3.8. The graph of the MC and SMC estimate function in Figure 3.8 shows the estimates of the Robustness function (3.5) as in Example 3.1 and is based on 20 samples of the Gaussian distribution. The curve depicts $\overset{smc}{R}(x_1, 0.3)$ and the line pieces between open (○) and closed (●) edges depicts $\overset{mc}{R}(x_1, 0.3)$.

Properties (3.17) and (3.18) imply that $d^{[in]}(a) + d^{[out]}(a) > 0$ and from Property (3.16) it follows that

$$s(a) = -\tfrac{1}{2\,\mathrm{M}} \tag{3.21}$$

Let $\mathbb{B}(a, r) = \left\{ x \in \mathbb{R}^{\mathrm{I}} \middle| \, \|a - x\| \leq r \right\}$ be a ball around $a$ with radius $r$.

[6] Let $\delta(x, v) = \min\limits_{w \in \partial\mathbb{H}(x)} \|v - w\|_2$ be the Euclidean distance between $v$ and the Happy set boundary. For point $v$ on the boundary, the distance to the boundary is $\delta(x, v) = 0 \Leftrightarrow g(x, v) = 0$. Functions $\delta$ and $g$ are both continuous in $x$. This means that if $g(x, v)$ converges to 0 for a sequence of $x$, then so does $\delta(x, v)$.

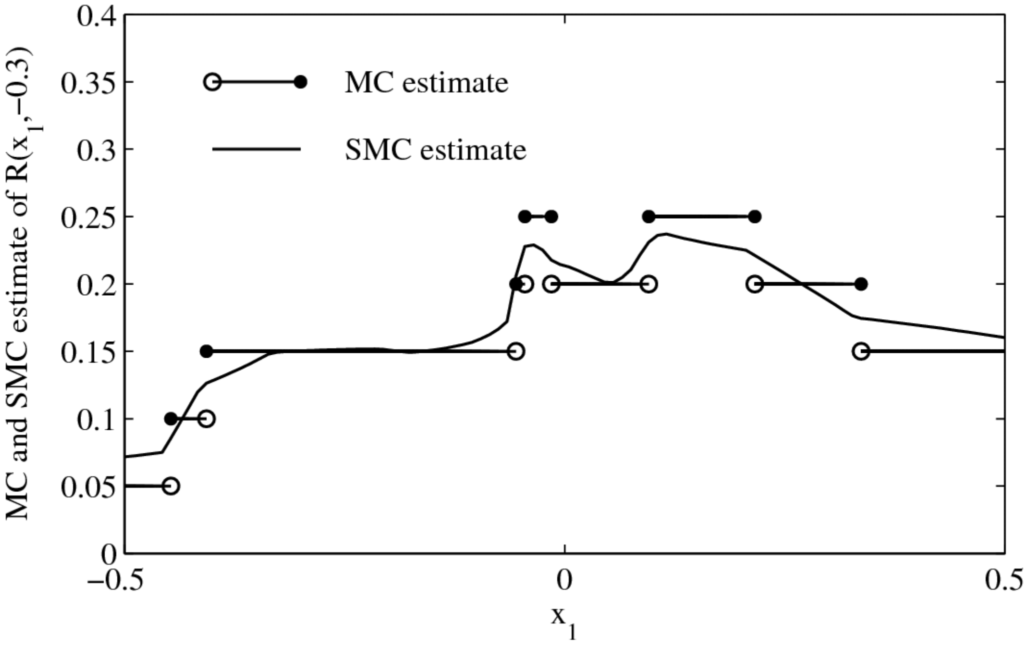


Figure 3.8: Continuous $\overset{\text{smc}}{R}(x_1, 0.3)$ and discontinuous (USC) $\overset{\text{mc}}{R}(x_1, 0.3)$

**Lemma 3.1** *Function $s$ is continuous in point $b \in \mathbb{S}$ if $\overset{mc}{R}$ is continuous in point $b$.*

**Proof.** $\overset{\text{mc}}{R}$ is continuous in point $b$, such that $\exists \delta > 0\ \forall x \in \mathbb{B}(b, \delta)$, $\overset{\text{mc}}{R}(x) = \overset{\text{mc}}{R}(b)$. Consequently, $\forall x \in \mathbb{B}(b, \delta)$, $\mathrm{M}\overset{\text{mc}}{R}(x) = |\left\{m | g(x, v^{[m]}) \geq 0\right\}| = |\left\{m | v^{[m]} \in \mathbb{H}(x)\right\}|$ is constant. For each $v^{[m]} \notin \mathbb{H}(b)$, $g(b, v^{[m]}) < 0$. Let $\mathbb{O}(b) = \left\{m | g(b, v^{[m]}) < 0\right\}$ be the set of indices of samples *outside* the Happy set at point $b$. Since function $g$ is continuous, $\exists \tilde{\delta} > 0, \tilde{\delta} < \delta$, such that $\forall m \in \mathbb{O}(b)$, $\forall x \in \mathbb{B}(b, \tilde{\delta})$: $g(x, v^{[m]}) < 0$. This means that $\forall x \in \mathbb{B}(b, \tilde{\delta})$, if $v^{[m]} \in \mathbb{H}(b)$ then $v^{[m]} \in \mathbb{H}(x)$ and if $v^{[m]} \notin \mathbb{H}(b)$ then $v^{[m]} \notin \mathbb{H}(x)$:

$$\forall x \in \mathbb{B}(b, \tilde{\delta}): \quad \begin{array}{l} \left\{m | v^{[m]} \in \mathbb{H}(x)\right\} = \left\{m | v^{[m]} \in \mathbb{H}(b)\right\} \text{ and} \\ \left\{m | v^{[m]} \notin \mathbb{H}(x)\right\} = \mathbb{O}(b) \end{array} \tag{3.22}$$

So, $\forall x \in \mathbb{B}(b, \tilde{\delta})$

$$\begin{aligned} d^{[in]}(x) &= \min_m \left\{ g\left(x, v^{[m]}\right) \middle| m \notin \mathbb{O}(b) \right\} \\ d^{[out]}(x) &= \min_m \left\{ -g\left(x, v^{[m]}\right) \middle| m \in \mathbb{O}(b) \right\} \end{aligned} \tag{3.23}$$

Since the minimum or maximum over a constant number of continuous functions is continuous, it follows that both $d^{[in]}$ and $d^{[out]}$ are continuous in point $b$. Because $\forall x \in \mathbb{S}$ $d^{[in]}(x) + d^{[out]}(x) > 0$, function $s$ is continuous in point $b$. ■

Now we arrive at the main property that tells us that the smoothing function $s(x)$ helps to make the $\overset{\text{smc}}{R}$ function continuous for the standard MC discontinuity points in $\mathbb{S}$.

The following reasoning is convenient. In a standard MC discontinuity point $a \in \mathbb{S}$, there is exactly one sample on the boundary of the Happy set. Let $v^{[m^*]}$ be this sample. Then $g(a, v^{[m^*]}) = 0$ and $g(a, v^{[m]}) \neq 0$ for all $m \neq m^*$. Since function $g(x, v)$ is continuous in $x$, Corollary 3.10 follows immediately.

**Corollary 3.10** *Let $a \in \mathbb{S}$ be a standard MC discontinuity point and sample $v^{[m^*]}$ the sample on the boundary of the Happy set. $\exists \overline{\delta}, \lambda > 0$ such that $\forall m \neq m^*$ $\forall y \in \mathbb{B}(a, \overline{\delta})$ $|g(y, v^{[m^*]})| < \lambda < |g(y, v^{[m]})|$.*

**Theorem 3.1** *The function $\overset{smc}{R}(x) = \overset{mc}{R}(x) + s(x)$ is continuous in each standard MC discontinuity point $a \in \mathbb{S}$.*

**Proof.** The core of the proof is to show that

$$\forall \varepsilon > 0 \; \exists \delta > 0 \text{ such that } x \in \mathbb{B}(a, \delta) \Longrightarrow \left| \overset{\mathrm{smc}}{R}(x) - \overset{\mathrm{smc}}{R}(a) \right| < \varepsilon \tag{3.24}$$

Corollary 3.10 shows that $\delta$ can be chosen such that the possibility that $d^{[out]}(x) = d^{[in]}(x)$ for $x \in \mathbb{B}(a, \delta)$ is excluded. Two situations can occur:

$$\begin{aligned} &\text{A: } x \in \mathbb{B}(a, \delta) \text{ and } d^{[in]}(x) < d^{[out]}(x) \\ &\text{B: } x \in \mathbb{B}(a, \delta) \text{ and } d^{[out]}(x) < d^{[in]}(x) \end{aligned}$$

The two situations are studied separately. First consider situation A: for $x \in \mathbb{B}(a, \delta)$ and $d^{[in]}(x) < d^{[out]}(x)$, where $v^{[m^*]}$ is the sample on the Happy set boundary, the following properties hold

$$g(x, v^{[m^*]}) \geq 0 \tag{3.25}$$

$$v^{[m^*]} \in \mathbb{H}(x) \tag{3.26}$$

$$d^{[in]}(x) = g(x, v^{[m^*]}) \tag{3.27}$$

$$d^{[out]}(x) > \lambda \tag{3.28}$$

$$-\tfrac{1}{2\mathrm{M}} \leq s(x) < 0 \tag{3.29}$$

$$\overset{\mathrm{mc}}{R}(a) - \overset{\mathrm{mc}}{R}(x) = 0 \tag{3.30}$$

$$\overset{\mathrm{smc}}{R}(a) - \overset{\mathrm{smc}}{R}(x) = -\tfrac{1}{2\mathrm{M}} - s(x) \tag{3.31}$$

Given $\varepsilon > 0$. According to Corollary 3.10 and the continuity of $g$, $\exists \delta > 0$ for which $\exists \lambda > 0$ such that $x \in \mathbb{B}(a, \delta) \Longrightarrow g(x, v^{[m^*]}) < \mathrm{M}\lambda\varepsilon$. It follows that

$$\begin{aligned} & -\tfrac{1}{2M} \leq s(x) = \tfrac{1}{2\mathrm{M}} \left( \tfrac{2d^{[in]}(x)}{d^{[in]}(x) + d^{[out]}(x)} - 1 \right) && < \tfrac{1}{2\mathrm{M}} \left( \tfrac{2g(x, v^{[m^*]})}{g(x, v^{[m^*]}) + \lambda} - 1 \right) \\ \Longrightarrow \quad & 0 \geq \overset{\mathrm{smc}}{R}(a) - \overset{\mathrm{smc}}{R}(x) && > -\tfrac{1}{\mathrm{M}} \left( \tfrac{g(x, v^{[m^*]})}{g(x, v^{[m^*]}) + \lambda} \right) \\ \Longrightarrow \quad & \left| \overset{\mathrm{smc}}{R}(x) - \overset{\mathrm{smc}}{R}(a) \right| < \tfrac{1}{\mathrm{M}} \left( \tfrac{g(x, v^{[m^*]})}{g(x, v^{[m^*]}) + \lambda} \right) && < \tfrac{g(x, v^{[m^*]})}{\mathrm{M}\lambda} < \varepsilon \end{aligned} \tag{3.32}$$

Situation B: for $x \in \mathbb{B}(a,\delta)$ and $d^{[out]}(x) < d^{[in]}(x)$ the following properties hold

$$g(x, v^{[m^*]}) < 0 \tag{3.33}$$

$$v^{[m^*]} \notin \mathbb{H}(x) \tag{3.34}$$

$$d^{[out]}(x) = -g(x, v^{[m^*]}) \tag{3.35}$$

$$d^{[in]}(x) > \lambda \tag{3.36}$$

$$0 \leq s(x) < \tfrac{1}{2\mathrm{M}} \tag{3.37}$$

$$\overset{\mathrm{mc}}{R}(a) - \overset{\mathrm{mc}}{R}(x) = \tfrac{1}{\mathrm{M}} \tag{3.38}$$

$$\overset{\mathrm{smc}}{R}(a) - \overset{\mathrm{smc}}{R}(x) = \tfrac{1}{2\mathrm{M}} - s(x) \tag{3.39}$$

Given $\varepsilon > 0$. According to Corollary 3.10 and the continuity of $g$, $\exists\delta > 0$ for which $\exists\lambda > 0$ such that $x \in \mathbb{B}(a,\delta) \implies -g(x, v^{[m^*]}) < \mathrm{M}\lambda\varepsilon$. It follows that

$$\begin{array}{llll} & \frac{1}{2\mathrm{M}} > s(x) = \frac{1}{2\mathrm{M}}\left(\frac{2d^{[in]}(x)}{d^{[in]}(x)+d^{[out]}(x)} - 1\right) & > & \frac{1}{2\mathrm{M}}\left(\frac{2\lambda}{\lambda - g(x,v^{[m^*]})} - 1\right) \\ \implies & 0 < \overset{\mathrm{smc}}{R}(a) - \overset{\mathrm{smc}}{R}(x) & < & \frac{1}{\mathrm{M}} - \frac{1}{\mathrm{M}}\left(\frac{\lambda}{\lambda - g(x,v^{[m^*]})}\right) \\ \implies & \left|\overset{\mathrm{smc}}{R}(x) - \overset{\mathrm{smc}}{R}(a)\right| & < & \frac{1}{\mathrm{M}} - \frac{1}{\mathrm{M}}\left(\frac{\lambda}{\lambda - g(x,v^{[m^*]})}\right) \\ \implies & \left|\overset{\mathrm{smc}}{R}(x) - \overset{\mathrm{smc}}{R}(a)\right| < \frac{1}{\mathrm{M}}\left(\frac{-g(x,v^{[m^*]})}{\lambda - g(x,v^{[m^*]})}\right) & < & \frac{-g(x,v^{[m^*]})}{\mathrm{M}\lambda} < \varepsilon \end{array} \tag{3.40}$$

■

The SMC method is elaborated in Example 3.4, based on Example 3.1 of Section 3.2.

**Example 3.4** *Consider the Robustness function $R(x)$ as defined in* (3.5) *of Example 3.1, where the random vector $\boldsymbol{v}$ is defined for two situations, respectively called the "Gaussian" situation and the "Exponential" situation. The mesh surface of the SMC estimate function in the Exponential and Gaussian situation, based on 20 samples, are shown in Figure 3.9. Note that the mesh surfaces in Figure 3.9 look more smooth than the mesh surfaces in Figure 3.1. The range of smoothing function s implies that $|\overset{smc}{R}(x) - \overset{mc}{R}(x)| \leq \frac{1}{2\mathrm{M}}$. For* M*=1000, the difference between the mesh surfaces of the SMC estimate and the MC estimate in 3.3 cannot be visually distinguished.*

*Optimisation of the SMC estimate function is run, similar to the Optimisation study of the MC estimate function of Example 3.1: The starting point is $x^{[0]} = (0.5, -0.3)$; FD step-sizes are $\Delta \in \left\{\left(\frac{1}{2}\right)^{10}, \left(\frac{1}{2}\right)^{9}, ..., \left(\frac{1}{2}\right)^{1}, \left(\frac{1}{2}\right)^{0}\right\}$; All other parameters are kept at the Matlab default values. The $\overset{smc}{R^*}$ column in Table 3.2 presents the values to which the SQP algorithm converges, for a given $\Delta$. For instance, the best result in the Gaussian situation with 20 samples was found with $\Delta = \left(\frac{1}{2}\right)^{10}$. The corresponding optimisation results are shown in Figure 3.10.*

Robustness maximisation based on the SMC function, yields better results than Robustness maximisation based on the MC estimate function, since most values of $\tilde{R}\,(x^*)$ in Table 3.2 are higher than the corresponding values $\tilde{R}\,(x^*)$ in Table 3.1. Note that using

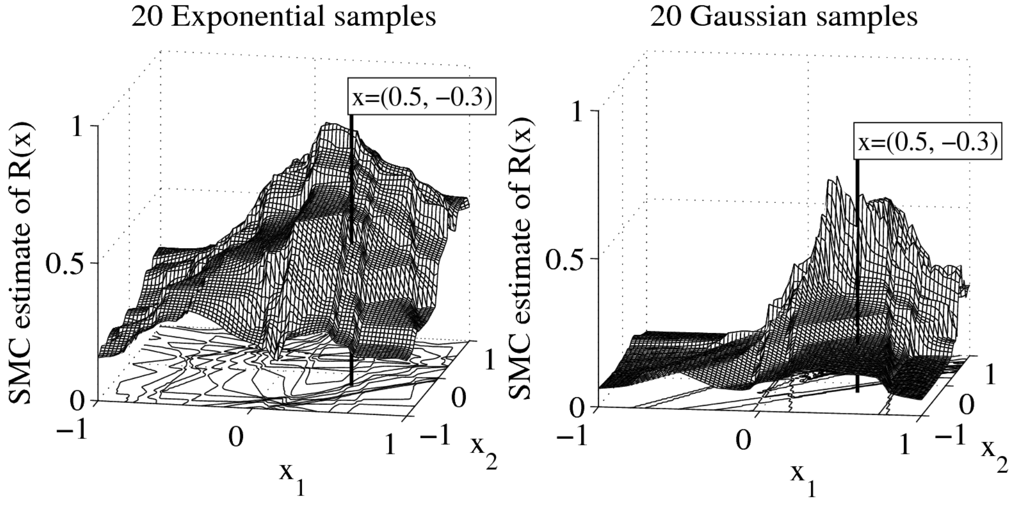


Figure 3.9: Illustration of SMC estimation in Exponential and Gaussian situation; 20 samples

the SMC function, $\tilde{R}\left(x^{*}\right)$ is higher than in the starting point $\tilde{R}\left(x^{[0]}\right)$ for all settings for the FD step-size $\Delta$. Since this is not the case in Example 3.1, we conclude that optimisation

Table 3.2: SQP Optimisation results of SMC estimate function, for a given $\Delta$, sample size and distribution type

| | Exponential case | | | | Gaussian case | | | |
|---|---|---|---|---|---|---|---|---|
| | M=20 | | M=1000 | | M=20 | | M=1000 | |
| $\Delta$ | $\overset{\text{smc}}{R^{*}}$ | $\tilde{R}\left(x^{*}\right)$ | $\overset{\text{smc}}{R^{*}}$ | $\tilde{R}\left(x^{*}\right)$ | $\overset{\text{smc}}{R^{*}}$ | $\tilde{R}\left(x^{*}\right)$ | $\overset{\text{smc}}{R^{*}}$ | $\tilde{R}\left(x^{*}\right)$ |
| Robustness estimate at starting point $x^{[0]}=(0.5,-0.3)$: | | | | | | | | |
| | 0.513 | 0.644 | 0.614 | 0.644 | 0.160 | 0.294 | 0.294 | 0.294 |
| $\left(\frac{1}{2}\right)^{10}$ | 0.835 | 0.940 | 0.948 | 0.954 | **0.653** | 0.763 | 0.743 | 0.731 |
| $\left(\frac{1}{2}\right)^{9}$ | 0.834 | 0.938 | **0.954** | 0.960 | 0.491 | 0.680 | **0.792** | 0.778 |
| $\left(\frac{1}{2}\right)^{8}$ | **0.858** | 0.960 | 0.954 | 0.959 | 0.625 | 0.769 | 0.784 | 0.770 |
| $\left(\frac{1}{2}\right)^{7}$ | 0.845 | 0.950 | 0.954 | 0.960 | 0.486 | 0.657 | 0.642 | 0.628 |
| $\left(\frac{1}{2}\right)^{6}$ | 0.813 | 0.935 | 0.954 | 0.960 | 0.489 | 0.673 | 0.791 | 0.775 |
| $\left(\frac{1}{2}\right)^{5}$ | 0.850 | 0.953 | 0.953 | 0.958 | 0.536 | 0.704 | 0.782 | 0.761 |
| $\left(\frac{1}{2}\right)^{4}$ | 0.830 | 0.938 | 0.951 | 0.956 | 0.317 | 0.429 | 0.785 | 0.764 |
| $\left(\frac{1}{2}\right)^{3}$ | 0.834 | 0.952 | 0.940 | 0.947 | 0.328 | 0.524 | 0.774 | 0.754 |
| $\left(\frac{1}{2}\right)^{2}$ | 0.742 | 0.917 | 0.915 | 0.925 | 0.240 | 0.330 | 0.716 | 0.702 |
| $\left(\frac{1}{2}\right)^{1}$ | 0.735 | 0.903 | 0.834 | 0.851 | 0.398 | 0.478 | 0.453 | 0.455 |
| $\left(\frac{1}{2}\right)^{0}$ | 0.513 | 0.643 | 0.681 | 0.671 | 0.239 | 0.329 | 0.296 | 0.292 |
| Average computation time and average estimated standard error per function evaluation are: | | | | | | | | |
| time (sec.) | 0.0035 | 50.258 | 0.1489 | 46.866 | 0.0036 | 152.513 | 0.1457 | 137.292 |
| $\hat{se}$ | 0.1068 | 0.0005 | 0.0143 | 0.0005 | 0.0744 | 0.0005 | 0.0129 | 0.0005 |

Comments on Table 3.2:

1. $\overset{\text{smc}}{R^{*}}$ is the Robustness estimate based on M samples, of point $x^{*}$ where SQP converged to.
2. $\tilde{R}(x^{*})$ is an MC estimate of $R(x^{*})$, with $\hat{se}\left(\tilde{\boldsymbol{R}}\left(x^{*}\right)\right) < 0.0005$ (See Appendix A.3).

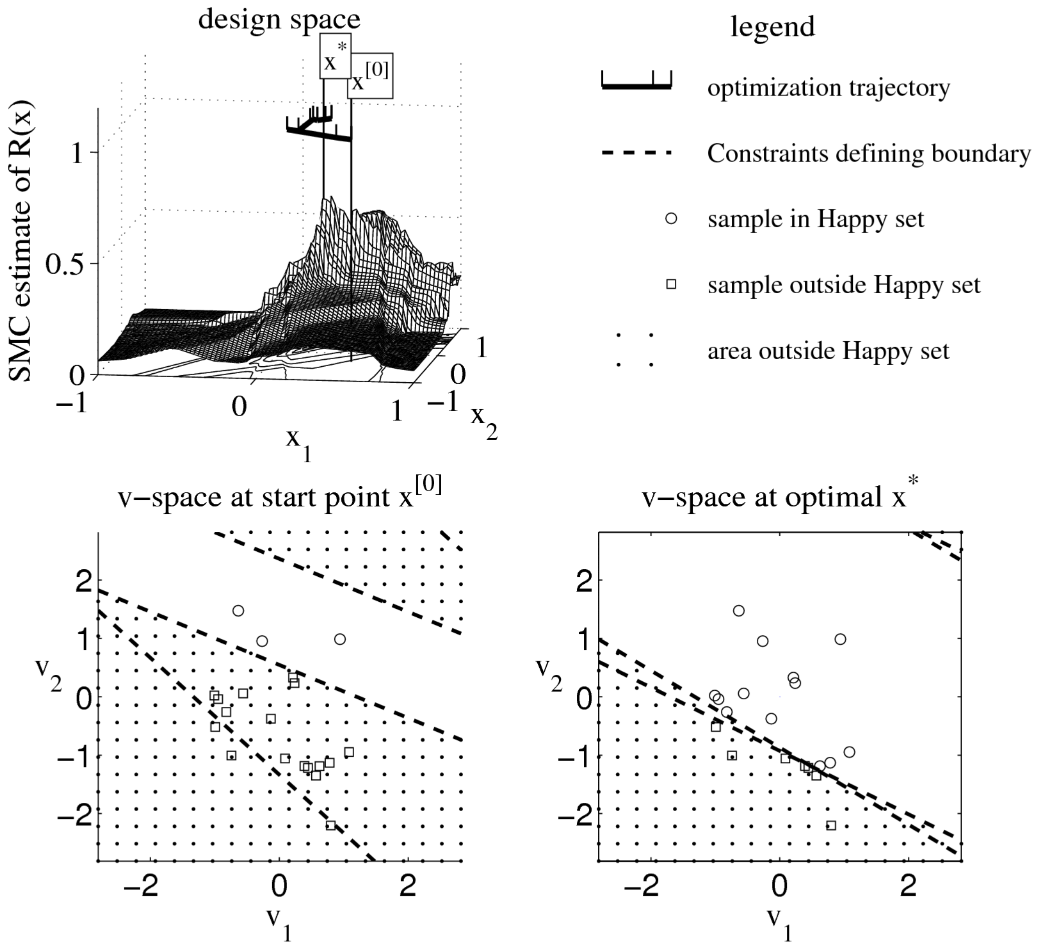


Figure 3.10: SMC estimate function optimisation results via SQP; Gaussian situation; M=20; $\Delta = \left(\frac{1}{2}\right)^{10}$; The unmarked white area between the dashed lines in the v-space depicts the Happy set

of the SMC function is less sensitive to the FD step-size setting.

## 3.5 N-1 Monte Carlo estimation method

The MC and SMC estimate functions are based on samples of $\boldsymbol{v}$, but do not use any other information about $\boldsymbol{v}$. In this section the N-1 Monte Carlo (N-1MC) estimate function is introduced for estimating the Robustness (2.2) or (2.3). The idea behind N-1 Monte Carlo estimation, is to use samples of $\boldsymbol{v}$ as well as the Probability Density Function ($f(v)$) information if $\boldsymbol{v}$ is a continuous random vector, or the probability measure ($\Pr_v$) information if $\boldsymbol{v}$ is a discrete random vector.

Let the random vector $\boldsymbol{v}$ consist of N elements and the MC method be based on M independent samples $v^{[m]} \in \mathbb{R}^{\mathrm{N}}$, $m = 1,..,\mathrm{M}$ of $\boldsymbol{v}$. Without loss of generality we consider the first element $v_1$ of $v$, separately from the other N-1 elements $v_2,..,v_{\mathrm{N}}$. For the discussion it is convenient to denote the components of the sample space as given in Corollary 2.5: $\mathbb{V} = \mathbb{V}_1 \times \mathbb{V}_2 \times \cdots \times \mathbb{V}_n \times \cdots \times \mathbb{V}_\mathrm{N}$ where $\mathbb{V}$ is the sample space of $\boldsymbol{v}$ and $\mathbb{V}_n$ is the sample space of each element $\boldsymbol{v}_n$. In Theorem 3.2 it is shown that the following estimate function is unbiased.

**Definition 3.7** *Let $q = (v_2, .., v_N)^\intercal \in \mathbb{R}^{\mathrm{N}-1}$ be a given realisation of the last N-1 elements of $\boldsymbol{v}$. The* N*-1MC estimate function ${}^{\textit{N-1mc}}r : \mathbb{R}^{\mathrm{N}-1} \times \mathbb{R}^\mathrm{I} \longrightarrow [0,1]$ is defined as*

$$ {}^{\textit{N-1mc}}r(q,x) = \Pr\left\{\boldsymbol{v}_1 \in \mathbb{V}_1 \,\middle|\, (\boldsymbol{v}_1, q)^\intercal \in \mathbb{H}(x)\right\} \tag{3.41} $$

**Corollary 3.11** *If $\boldsymbol{v}$ is a continuous random vector, then*

$$ {}^{\textit{N-1mc}}r(q,x) = \int\limits_{\mathbb{V}_1} I(v_1, q, x) f_1(v_1) dv_1 \tag{3.42} $$

*with $f_1(v_1)$ the PDF of the first element of $\boldsymbol{v}$ and $I$ the Happy set indicator function as defined in* (2.5)*.*

**Corollary 3.12** *If $\boldsymbol{v}$ is a discrete random vector, then*

$$ {}^{\textit{N-1mc}}r(q,x) = \sum_{v_1 \in \mathbb{V}_1} I(v_1, q, x) \Pr\{\boldsymbol{v}_1 = v_1\} \tag{3.43} $$

*with $I$ the Happy set indicator function as defined in* (2.5)*.*

**Theorem 3.2** *Let ${}^{\textit{N-1mc}}r(q,x)$ be as defined in 3.7 and $\boldsymbol{q} = [\boldsymbol{v}_2, .., \boldsymbol{v}_N]$ be the corresponding random vector.*

$$ E_{\boldsymbol{q}}[{}^{\textit{N-1mc}}r(\boldsymbol{q}, x)] = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} \tag{3.44} $$

**Proof.** First, consider the case that $\boldsymbol{v}$ is a continuous random vector. Since

$$ f_1(v_1)dv_1 f_n(v_n)dv_n = f_1(v_1)f_n(v_n)dv_1dv_n \text{ for } n = 2, .., \mathrm{N} $$

the terms can be rearranged as follows:

$$ \begin{aligned} & E_{\boldsymbol{q}}[{}^{\text{N-1mc}}r(\boldsymbol{v}_2, .., \boldsymbol{v}_\mathrm{N}, x)] \\ =& \int\limits_{\mathbb{V}_2} \cdots \int\limits_{\mathbb{V}_\mathrm{N}} {}^{\text{N-1mc}}r(v_2, .., v_\mathrm{N}, x) f_2(v_2) \cdots f_\mathrm{N}(v_\mathrm{N}) dv_2 \cdots dv_\mathrm{N} \\ =& \int\limits_{\mathbb{V}_2} \cdots \int\limits_{\mathbb{V}_\mathrm{N}} \int\limits_{\mathbb{V}_1} I(v_1, v_2, .., v_\mathrm{N}, x) f_1(v_1) dv_1 f_2(v_2) \cdots f_\mathrm{N}(v_\mathrm{N}) dv_2 \cdots dv_\mathrm{N} \\ =& \int\limits_{\mathbb{V}_1} \cdots \int\limits_{\mathbb{V}_\mathrm{N}} I(v_1, .., v_\mathrm{N}, x) f_1(v_1) \cdots f_\mathrm{N}(v_\mathrm{N}) dv_1 \cdots dv_\mathrm{N} \\ =& \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} \end{aligned} \tag{3.45} $$

Finally, consider the case that $\boldsymbol{v}$ is a discrete random vector.

$$\begin{aligned}
&E_{\boldsymbol{q}}[\overset{\text{N-1mc}}{r}(\boldsymbol{v}_2,..,\boldsymbol{v}_{\mathrm{N}},x)]\\
&=\sum_{v_2\in\mathbb{V}_2}\sum_{v_3\in\mathbb{V}_3}\cdots\sum_{v_{\mathrm{N}}\in\mathbb{V}_{\mathrm{N}}}\overset{\text{N-1mc}}{r}(v_2,v_3,..,v_{\mathrm{N}},x)\mathrm{Pr}\left\{\boldsymbol{v}_2=v_2\right\}\mathrm{Pr}\left\{\boldsymbol{v}_3=v_3\right\}\cdots\mathrm{Pr}\left\{\boldsymbol{v}_{\mathrm{N}}=v_{\mathrm{N}}\right\}\\
&=\sum_{v_1\in\mathbb{V}_1}\sum_{v_2\in\mathbb{V}_2}\cdots\sum_{v_{\mathrm{N}}\in\mathbb{V}_{\mathrm{N}}}I(v_1,v_2..,v_{\mathrm{N}},x)\mathrm{Pr}\left\{\boldsymbol{v}_1=v_1\right\}\mathrm{Pr}\left\{\boldsymbol{v}_2=v_2\right\}\cdots\mathrm{Pr}\left\{\boldsymbol{v}_{\mathrm{N}}=v_{\mathrm{N}}\right\}\\
&=\sum_{v\in\mathbb{V}}I(v,x)\,\mathrm{Pr}\left\{\boldsymbol{v}=v\right\}=\mathrm{Pr}\left\{\boldsymbol{v}\in\mathbb{H}(x)\right\}
\end{aligned}\tag{3.46}$$

■

Consider $\boldsymbol{q}^{[m]}\sim\boldsymbol{q}$ with $m=1,..,\mathrm{M}$. Theorem 3.2 leads to:

**Corollary 3.13**
*The estimator*

$$\overset{\text{N-1mc}}{\boldsymbol{R}}(x)=\frac{1}{M}\sum_{m=1}^{M}\overset{\textit{N-1mc}}{r}\left(\boldsymbol{q}^{[m]},x\right)\tag{3.47}$$

*is an unbiased estimator of $R(x)$*

**Corollary 3.14** *Following the definitions of Rice (1995), the standard error of the estimator is*

$$se\left(\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right)=\sqrt{\frac{1}{\mathrm{M}}\left(E\left[\left(\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right)^2\right]-\left(E\left[\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right]\right)^2\right)}\tag{3.48}$$

**Theorem 3.3**

$$se\left(\overset{\textit{N-1mc}}{\boldsymbol{R}}(x)\right)\leq se\left(\overset{\textit{mc}}{\boldsymbol{R}}(x)\right)\tag{3.49}$$

**Proof.**
It is sufficient to prove the equation for $\mathrm{M}=1$, since the standard error of both estimators decreases equally proportional with $\sqrt{\mathrm{M}}$. As each realisation $\overset{\text{N-1mc}}{R}(x)\leq 1$ and $\overset{\text{mc}}{R}(x)\in\{0,1\}$, it holds that $0\leq\left(\overset{\text{N-1mc}}{R}(x)\right)^2\leq\overset{\text{N-1mc}}{R}(x)\leq 1$ and $\left(\overset{\text{mc}}{R}(x)\right)^2=\overset{\text{mc}}{R}(x)$. Therefore,

$$\begin{aligned}
var\left(\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right)&=E\left[\left(\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right)^2\right]&&-\left(E\left[\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right]\right)^2\\
&\leq E\left[\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right]&&-\left(E\left[\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right]\right)^2\\
&=E\left[\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)^2\right]&&-\left(E\left[\overset{\text{mc}}{\boldsymbol{R}}(x)\right]\right)^2=var\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)
\end{aligned}$$

For general M, from independency and unbiasedness of both estimators, follows now

$$se\left(\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right)=\left[VAR(\overset{\text{N-1mc}}{\boldsymbol{R}}(x))\right]^{1/2}\leq\left[VAR(\overset{\text{mc}}{\boldsymbol{R}}(x))\right]^{1/2}=se\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)\tag{3.50}$$

■

The standard error of the MC estimator and the standard error of the N-1MC estimator can be equal. This happens when the Happy set has the following characteristic.

**Definition 3.8** *Let* $\overset{\textit{N-1mc}}{r}(q,x)$ *be defined by* (3.41) *and* $\boldsymbol{q} = [\boldsymbol{v}_2, .., \boldsymbol{v}_N]$ *the corresponding random vector. The Happy set* $\mathbb{H}(x)$ *is called to have an* All-Or-Nothing *shape if* $\Pr\left\{\overset{\textit{N-1mc}}{r}(\boldsymbol{q},x) \in \{0,1\}\right\} = 1$.

If the Happy set has an All-Or-Nothing shape, then $\overset{\text{N-1mc}}{r}(\boldsymbol{q},x)$ is *almost surely* either 0 or 1. Recall that the range of the MC estimate function for M=1 is also $\{0,1\}$. Theorem 3.4 shows that these range properties lead to identical standard errors.

**Theorem 3.4** *For a measurable set* $\mathbb{H}(x)$ *holds that*

$$se\left(\overset{\textit{N-1mc}}{\boldsymbol{R}}(x)\right) = se\left(\overset{\textit{mc}}{\boldsymbol{R}}(x)\right)$$

*if and only if* $\mathbb{H}(x)$ *has an All-Or-Nothing shape.*

**Proof.**

Let $\mathbb{V}^{\mathrm{N}-1} = \mathbb{V}_2 \times \mathbb{V}_3 \times \cdots \times \mathbb{V}_n \times \cdots \times \mathbb{V}_{\mathrm{N}}$. In the proof of Theorem 3.3, the equality

$$E\left[\left(\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right)^2\right] = E\left[\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)^2\right]$$

is equivalent to

$$E\left[\left(\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right)^2\right] = E\left[\overset{\text{N-1mc}}{\boldsymbol{R}}(x)\right]$$

which holds if and only if any realisation of the estimator $\overset{\text{N-1mc}}{\boldsymbol{R}}(x)$, with M= 1, is either 0 or 1, i.e. if and only if the set

$$\left\{q \in \mathbb{V}^{\mathrm{N}-1} \,\middle|\, \overset{\text{N-1mc}}{r}(q,x) \in \{0,1\}\right\}$$

is dense in $\mathbb{V}^{\mathrm{N}-1}$, i.e. is equal to $\mathbb{V}^{\mathrm{N}-1}$ apart from a set of measure zero. ∎

From theorem 3.4 follows directly

**Corollary 3.15**
*Given the Happy set* $\mathbb{H}(x)$

$$se\left(\overset{\textit{N-1mc}}{\boldsymbol{R}}(x)\right) < se\left(\overset{\textit{mc}}{\boldsymbol{R}}(x)\right)$$

*if and only if* $\mathbb{H}(x)$ *does not have an All-Or-Nothing shape.*

Notice that a bounded Happy set $\mathbb{H}(x)$ with non-empty interior does not have an All-Or-Nothing shape, if $\mathbb{V} = \mathbb{R}^{\mathrm{N}}$, i.e. $\boldsymbol{v}$ has a continuous distribution on $\mathbb{R}^{\mathrm{N}}$. Consequently, if the Happy set has these properties, then the N-1MC estimator is more accurate than the MC estimator.

Let us focus on the computation of the N-1MC estimate function. From a practical point of view, it is convenient if the Happy set leads to

$$\begin{aligned} a &= \inf\{v_1 \in \mathbb{V}_1 | (v_1, q_1, .., q_{\mathrm{N}-1}) \in \mathbb{H}(x)\} \\ b &= \sup\{v_1 \in \mathbb{V}_1 | (v_1, q_1, .., q_{\mathrm{N}-1}) \in \mathbb{H}(x)\} \\ a < t < b &\Longrightarrow t \in \{v_1 \in \mathbb{V}_1 | (v_1, q_1, .., q_{\mathrm{N}-1}) \in \mathbb{H}(x)\} \end{aligned}$$

for any $q \in \mathbb{V}^{\mathrm{N}-1}$, because then, for a continuous random variable $\boldsymbol{v}_1$

$$\overset{\text{N-1mc}}{r}(q, x) = \int_a^b f_1(v_1) dv_1 = F_1(b) - F_1(a)$$

or a discrete random variable $\boldsymbol{v}_1$

$$\overset{\text{N-1mc}}{r}(q, x) = \sum_{a<v_1\leq b} f_1(v_1) dv_1 = F_1(b) - F_1(a)$$

This is convenient, because a computer program such as the Statistics Toolbox of Matlab®, can be used to compute the Cumulative Distribution Function (CDF) $F_1(x) = \Pr\{\boldsymbol{v}_1 \leq x\}$, of discrete or continuous distributed random variable $\boldsymbol{v}_1$. If the Happy set has the following characteristic, then it is possible to compute (3.42) or (3.43), using the CDF of $\boldsymbol{v}_1$.

**Definition 3.9** *The measurable set* $\mathbb{H}(x)$ *is called* Dim-1-Convex *if* $\{v_1 \in \mathbb{V}_1 \, | (v_1, q_1, .., q_{N-1}) \in \mathbb{H}(x)\}$ *is convex for all* $q \in \mathbb{V}^{\mathrm{N}-1}$.

Note that Definition 3.9 implies that any convex Happy set is also Dim-1-Convex. The converse is not true, i.e. a Dim-1-convex set is not necessarily a convex set. Figure 3.11 illustrates a non-convex set, which is Dim-1-Convex with respect to $v_1$with $q = v_2$. Note that the set illustrated in Figure 3.11 is not Dim-1-Convex with respect to $v_2$ with $q = v_1$.

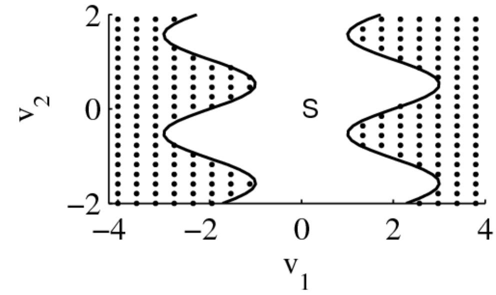
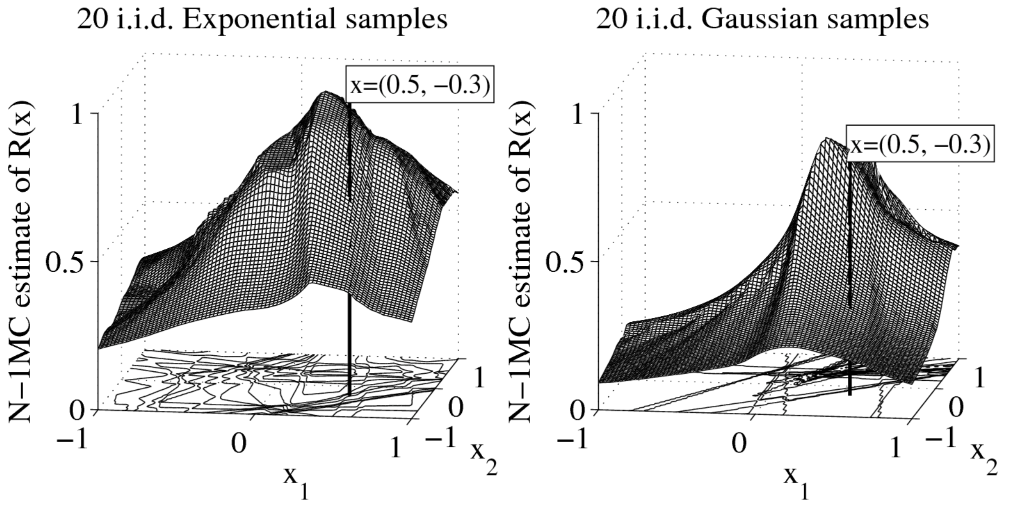


Figure 3.11: Illustration of a non-convex set S that is Dim-1-Convex for $q = v_2$

The idea of N-1 MC Robustness computation, for Dim-1-Convex sets, goes as follows. The Robustness estimate can be computed conveniently, for M independent samples $q^{[1]}, .., q^{[m]}, .., q^{[\mathrm{M}]}$ of random vector $\boldsymbol{q}$, with

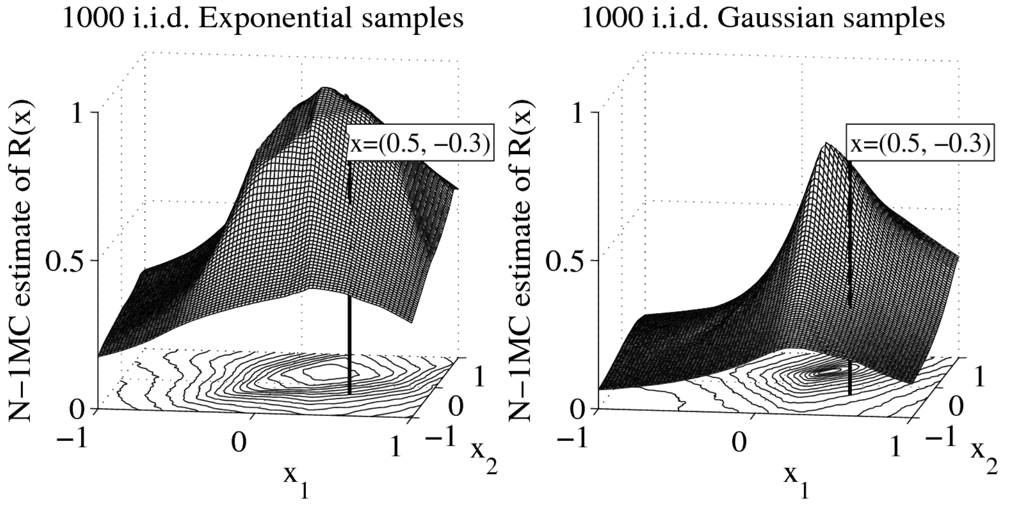
$$\overset{\textit{N-1mc}}{R}(x) = \tfrac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} \overset{\text{N-1mc}}{r}(q^{[m]}, x) = \tfrac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} \left[F_1\left(b(q^{[m]}, x)\right) - F_1\left(a(q^{[m]}, x)\right)\right] \qquad (3.51)$$



20 i.i.d. Exponential samples

20 i.i.d. Gaussian samples

Figure 3.12: Illustration of N-1MC estimate function; in Exponential and Gaussian case; 20 samples



1000 i.i.d. Exponential samples

1000 i.i.d. Gaussian samples

Figure 3.13: Illustration of N-1MC estimate function; Exponential and Gaussian case; 1000 samples

where

$$\begin{aligned} a(q,x) &= \inf\left\{v_1 \in \mathbb{V}_1 \,|(v_1, q_1, .., q_{\mathrm{N}-1}) \in \mathbb{H}(x)\right\} \\ b(q,x) &= \sup\left\{v_1 \in \mathbb{V}_1 \,|(v_1, q_2, .., q_{\mathrm{N}-1}) \in \mathbb{H}(x)\right\} \end{aligned} \tag{3.52}$$

with standard error estimate

$$\hat{se}\left(\overset{\textit{N-1mc}}{\boldsymbol{R}}(x)\right) = \sqrt{\frac{1}{\mathrm{M}-1}\left(\tfrac{1}{\mathrm{M}}\sum_{m=1}^{\mathrm{M}} \overset{\text{N-1mc}}{r}(q^{[m]},x)^2 - \left(\overset{\textit{N-1mc}}{R}(x)\right)^2\right)} \tag{3.53}$$

Note that if the Happy set $\mathbb{H}(x)$ is a finite union $\bigcup_{h=1}^{\mathrm{H}} \mathbb{H}^{[h]}(x)$ of H disjunct Dim-1-Convex sets $\mathbb{H}^{[h]}(x)$, then the probability estimator of $\mathbb{H}(x)$ is the sum of the N-1MC estimators of the individual sets.

The N-1MC estimate function (3.51) has an analytic expression for the gradient, if $\frac{\partial a(q^{[m]},x)}{\partial x_i}$ and $\frac{\partial b(q^{[m]},x)}{\partial x_i}$ are defined:

$$\frac{\partial \overset{N\text{-}1mc}{R}(x)}{\partial x_i} = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} f_1\left(b(q^{[m]},x)\right) \frac{\partial b(q^{[m]},x)}{\partial x_i} - f_1\left(a(q^{[m]},x)\right) \frac{\partial a(q^{[m]},x)}{\partial x_i},\ i = 1,..,\mathrm{I} \qquad (3.54)$$

If M is relatively large, or if $\frac{\partial a(q^{[m]},x)}{\partial x_i}$ and $\frac{\partial b(q^{[m]},x)}{\partial x_i}$ are difficult to compute, then finite differencing is a more practical alternative.

The optimisation of $\overset{N\text{-}1mc}{\boldsymbol{R}}(x)$ is illustrated for an extension of Example 3.1.

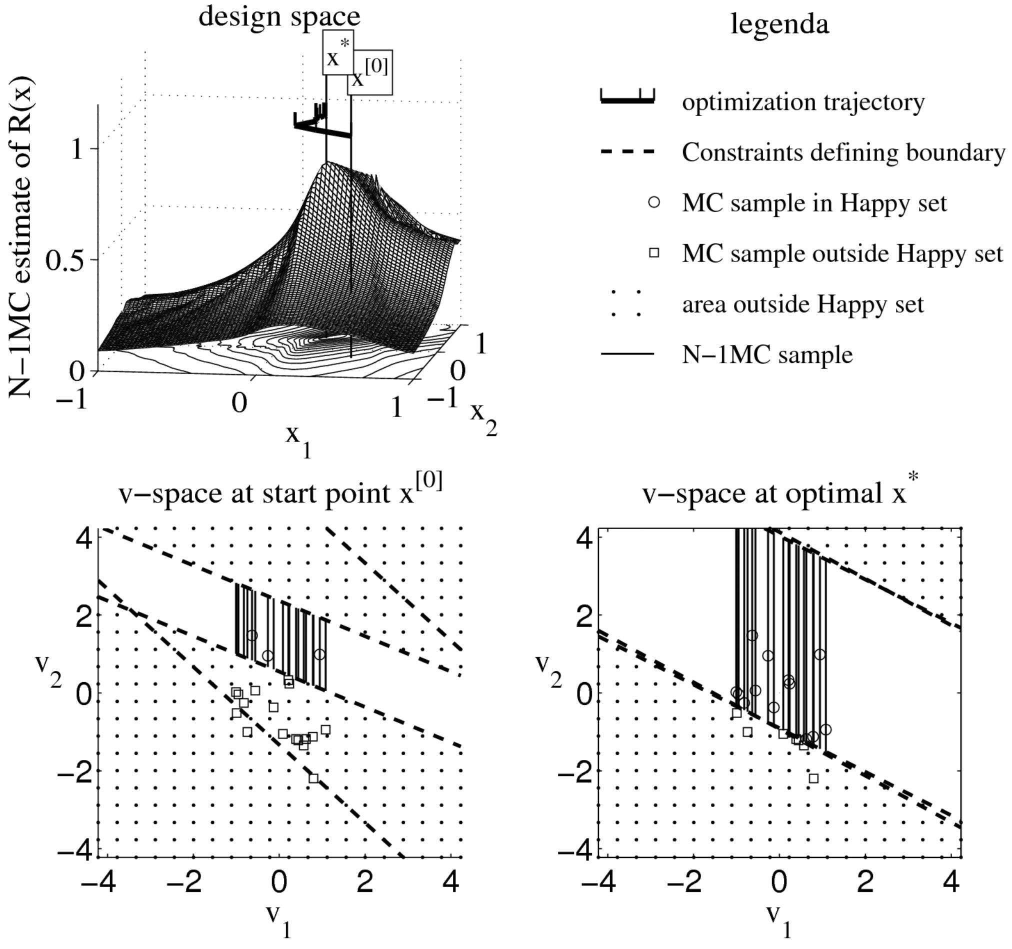


Figure 3.14: SQP optimisation results of N-1MC estimate function; Gaussian case; M=20; $\Delta = \left(\frac{1}{2}\right)^7$

**Example 3.5** *In Example 3.1, any dimension in the v-space can be chosen as an integration direction in the N-1MC method. In this example, the N-1MC estimate function*

Table 3.3: SQP Optimisation results of N-1MC estimate function, for a given Δ, sample size and distribution type

| | Exponential case | | | | Gaussian case | | | |
|---|---|---|---|---|---|---|---|---|
| | M=20 | | M=1000 | | M=20 | | M=1000 | |
| Δ | $\overset{\text{N-1mc}}{R^*}$ | $\tilde{R}(x^*)$ | $\overset{\text{N-1mc}}{R^*}$ | $\tilde{R}(x^*)$ | $\overset{\text{N-1mc}}{R^*}$ | $\tilde{R}(x^*)$ | $\overset{\text{N-1mc}}{R^*}$ | $\tilde{R}(x^*)$ |
| Robustness estimate at starting point $x^{[0]} = (0.5, -0.3)$: | | | | | | | | |
| | 0.649 | 0.642 | 0.638 | 0.642 | 0.290 | 0.293 | 0.288 | 0.293 |
| $\left(\frac{1}{2}\right)^{10}$ | 0.952 | 0.959 | **0.959** | **0.960** | 0.800 | 0.782 | **0.781** | **0.783** |
| $\left(\frac{1}{2}\right)^{9}$ | 0.950 | 0.958 | 0.958 | 0.958 | 0.797 | 0.778 | 0.778 | 0.781 |
| $\left(\frac{1}{2}\right)^{8}$ | 0.948 | 0.957 | 0.959 | 0.960 | 0.790 | 0.768 | 0.778 | 0.780 |
| $\left(\frac{1}{2}\right)^{7}$ | 0.950 | 0.957 | 0.959 | 0.959 | **0.801** | **0.783** | 0.781 | 0.783 |
| $\left(\frac{1}{2}\right)^{6}$ | **0.952** | **0.959** | 0.957 | 0.958 | 0.795 | 0.777 | 0.764 | 0.766 |
| $\left(\frac{1}{2}\right)^{5}$ | 0.949 | 0.956 | 0.954 | 0.956 | 0.801 | 0.783 | 0.764 | 0.767 |
| $\left(\frac{1}{2}\right)^{4}$ | 0.938 | 0.950 | 0.955 | 0.956 | 0.752 | 0.709 | 0.763 | 0.765 |
| $\left(\frac{1}{2}\right)^{3}$ | 0.941 | 0.948 | 0.954 | 0.955 | 0.706 | 0.673 | 0.775 | 0.778 |
| $\left(\frac{1}{2}\right)^{2}$ | 0.935 | 0.942 | 0.919 | 0.926 | 0.628 | 0.596 | 0.622 | 0.625 |
| $\left(\frac{1}{2}\right)^{1}$ | 0.829 | 0.896 | 0.843 | 0.849 | 0.450 | 0.462 | 0.497 | 0.501 |
| $\left(\frac{1}{2}\right)^{0}$ | 0.650 | 0.646 | 0.638 | 0.644 | 0.380 | 0.329 | 0.369 | 0.356 |
| Average computation time and average estimated standard error per function evaluation are: | | | | | | | | |
| time (sec.) | 0.0074 | 45.580 | 0.3453 | 47.813 | 0.0068 | 136.696 | 0.3097 | 131.480 |
| $\hat{se}$ | 0.0602 | 0.0005 | 0.0076 | 0.0005 | 0.0444 | 0.0005 | 0.0065 | 0.0005 |

Comments on Table 3.3:

1. $\overset{\text{N-1mc}}{R^*}$ is the Robustness estimate based on M samples, of point $x^*$ where SQP converged to.
2. $\tilde{R}(x^*)$ is an MC estimate of $R(x^*)$, with $\hat{se}\left(\tilde{\boldsymbol{R}}(x^*)\right) < 0.0005$ (See Appendix A.3).

*evaluations are based on integrating over $v_2$. The resulting mesh surface of the N-1MC estimate function in the Exponential and Gaussian situation, based on 20 samples, are shown in Figure 3.12.*

*The mesh surfaces in Figure 3.12 look more smooth than the mesh surfaces of respectively the MC method in Figure 3.1 and the SMC method in Figure 3.9. The standard error of the N-1MC in Table 3.3, relative to the standard error of the MC method in Table 3.1, shows that the N-1MC method is almost twice as accurate, given the same number of samples. This observation supports the idea that a more accurate estimate, results into a more smooth surface (under the assumption that $R(x)$ is smooth). Likewise, Figure 3.13 shows the N-1MC estimate of (3.5) as a function of x, based on 1000 samples.*

*Optimisation of the N-1MC estimate function is run and compared to the Optimisation of the MC estimate function of Example 3.3: The starting point is $x^{[0]} = (0.5, -0.3)$; FD step-sizes are $\Delta \in \left\{\left(\frac{1}{2}\right)^{10}, \left(\frac{1}{2}\right)^{9}, ..., \left(\frac{1}{2}\right)^{1}, \left(\frac{1}{2}\right)^{0}\right\}$; All other parameters are kept at Matlab default values. Table 3.3 gives the values of $\overset{N-1mc}{R^*}$ to which the SQP algorithm converges. Table 3.3 shows that N-1MC yields better results than MC. However N-1MC and SMC appear to be competitive for this example. For instance, the best result in the Gaussian situation with 20 samples was found with $\Delta = \left(\frac{1}{2}\right)^{7}$. The corresponding optimisation results are shown in Figure 3.14. The vertical line pieces in Figure 3.14, illustrate the* *intervals in the $v_2$-space over which the PDF of $\boldsymbol{v}_2$ is integrated, The first element of the sample values give the horizontal position of these line pieces, i.e. $q^{[m]} = v_1^{[m]}$ for $m = 1, ..,$M.*

## 3.6 Directional Sampling estimation method

In this section, the focus is on estimating the Robustness (2.4) for the situation that $\boldsymbol{v}$ is a continuous random vector which is characterised by a so-called *spherical symmetric* PDF. Spherical symmetry of a function is defined as follows.

**Definition 3.10** *A function $f : \mathbb{R}^{\mathrm{N}} \longrightarrow \mathbb{R}$ is spherical symmetric if and only if $\exists g : \mathbb{R} \longrightarrow \mathbb{R}$, such that $\forall v \in \mathbb{R}^{\mathrm{N}}$ $g\left(v^{\intercal}v\right) = f(v)$*

A spherical symmetric function $f : \mathbb{R}^{\mathrm{N}} \longrightarrow \mathbb{R}$ has the property $f(a) = f(b)$, for two points $a \in \mathbb{R}^{\mathrm{N}}$ and $b \in \mathbb{R}^{\mathrm{N}}$ on the same sphere, i.e. $\|a\|_2 = \|b\|_2$. A typical example of such a distribution is the standard Multivariate Normal distribution (MVN), defined by the PDF $\varphi(v) = \frac{1}{(2\pi)^{\mathrm{N}/2}} e^{-\frac{1}{2}v^{\intercal}v}$.

The MVN distribution plays an important role in Probability Theory, Mathematical Statistics and Stochastic Programming. Deák (2003, 2000) states that the computation of the probability that a MVN distributed random vector has a realisation in an N dimensional set, is frequently required in diverse areas of computational mathematics. For example, to determine the probability of breakdowns in problems of structural reliability, or the probability of multidimensional confidence intervals, or the probability of feasibility regions in Stochastic Programming. Deák's method is also called Deák's Decomposition (in Gassmann et al., 2002) or Directional Sampling (in Bjerager, 1988).

Numerical integration of (2.7) when $f(v) = \varphi(v)$ is not tractable in general, which motivated Genz (1993), Szántai (2000), Deák (2003) and Somerville (1998) to develop alternative methods for estimating (2.7). In the last decades, specific methods have been studied for the situation that $\mathbb{H}$ has an *explicit shape*: Gassmann et al. (2002) discusses methods for the special case that $\mathbb{H}$ has a rectangle shape. In (Deák, 2003, 2000) methods are discussed for the case that the shape of $\mathbb{H}$ is a polyhedron, a hyper-ellipsoid, or a circular cone. Note that these shapes can be described by a system of linear and quadratic inequalities.

The main contribution of this section is the generalisation of Deák's method. The generalisation of Deák's method is called the Directional Sampling (DS) estimation method. Deák's method is generalised in three ways. Firstly, Deák's method can be generalised for any continuous random vector with a so-called spherical symmetric PDF. Secondly, the basic principles of Deák's method hold for any measurable set $\mathbb{H}(x)$. Thirdly, the underlying properties of the Happy set are defined, that imply the possibility to compute the DS estimate numerically and guarantee that the DS estimator is more accurate than the MC estimator.

The focus of this section, given a spherical symmetric PDF $f(v) = g\left(v^{\intercal} v\right)$, is on estimating the Robustness

$$R(x) = \int\limits_{\mathbb{H}(x)} g\left(v^{\intercal} v\right) dv \tag{3.55}$$

The main idea in Deák's method, is to transform expression (3.55) into a radial and a spherical part via a Polar Coordinates Transform, leading to an efficient unbiased estimator $\overset{\text{ds}}{\boldsymbol{R}}(x)$ for (3.55).

To derive this efficient unbiased estimator, it is convenient first to give an explicit definition of the Polar Coordinates Transform in Section 3.6.1, followed by the introduction of a key-concept called the *random direction* in Section 3.6.2, before defining $\overset{\text{ds}}{\boldsymbol{R}}(x)$ in Section 3.6.3.

## 3.6.1 Polar Coordinates Transform

The following Polar Coordinates Transform is based on definitions in Prékopa (2001).

**Definition 3.11** *The Polar Coordinates Transform for* N$\geq$ 2 *is defined by the following mapping:*

$$P : \mathbb{P} \longrightarrow \mathbb{R}^{\mathrm{N}} \tag{3.56}$$

$$P \begin{pmatrix} r \\ \alpha_1 \\ \vdots \\ \alpha_{\mathrm{N}-1} \end{pmatrix} = \begin{pmatrix} r\sin(\alpha_1) \\ r\cos(\alpha_1)\sin(\alpha_2) \\ \vdots \\ r\cos(\alpha_1)\cdots\cos(\alpha_{\mathrm{N}-3})\sin(\alpha_{\mathrm{N}-2}) \\ r\cos(\alpha_1)\cdots\cos(\alpha_{\mathrm{N}-3})\cos(\alpha_{\mathrm{N}-2})\sin(\alpha_{\mathrm{N}-1}) \\ r\cos(\alpha_1)\cdots\cos(\alpha_{\mathrm{N}-3})\cos(\alpha_{\mathrm{N}-2})\cos(\alpha_{\mathrm{N}-1}) \end{pmatrix} \tag{3.57}$$

*where* $\mathbb{P} = \mathbb{R}_+ \times \mathbb{A}$ *with* $\mathbb{A} = \left\{\alpha \in \mathbb{R}^{\mathrm{N}-1} \middle| \, |\alpha_i| \leq \frac{1}{2}\pi, i = 1, .., \mathrm{N}-2; |\alpha_{\mathrm{N}-1}| \leq \pi\right\}$

**Corollary 3.16** *The Jacobian determinant of the Polar Coordinates Transform is* $J : \mathbb{P} \longrightarrow \mathbb{R}$ *for* $p \in \mathbb{P}$ *with*

$$J(p) = det\frac{\partial P(p)}{\partial p} = r^{\mathrm{N}-1} K(\alpha) \tag{3.58}$$

*where* $K : \mathbb{A} \longrightarrow \mathbb{R}_+$ *with* $K(\alpha) = \begin{cases} \cos^{\mathrm{N}-2}(\alpha_1)\cos^{\mathrm{N}-3}(\alpha_2)\cdots\cos(\alpha_{N-2}) & \textit{if } \mathrm{N} > 4 \\ \cos^{\mathrm{N}-2}(\alpha_1)\cos^{\mathrm{N}-3}(\alpha_2) & \textit{if } \mathrm{N} = 4 \\ \cos^{\mathrm{N}-2}(\alpha_1) & \textit{if } \mathrm{N} = 3 \\ 1 & \textit{if } \mathrm{N} = 2 \end{cases}$

The derivation of the Jacobian of the Polar Coordinates Transform is given among others in Anderson (1984).

### 3.6.2 Random Direction

As mentioned in the beginning of Section 3.6, the approach is to redefine (3.55) in a spherical and a radial part. The concept of a random direction is closely related to the spherical part discussed later.

**Definition 3.12** *Let* $\mathbb{S} = \left\{ s \in \mathbb{R}^{\mathrm{N}} \middle| \, s's = 1 \right\}$ *be the unit hyper-sphere. A random direction* $\boldsymbol{s}$ *is a random vector, uniformly distributed on the unit hyper-sphere and is defined on probability space*[7] $(\mathbb{S}, \mathcal{S}, \mathrm{Pr}_{\boldsymbol{s}})$.

**Corollary 3.17** *Let* $A : \mathbb{S} \longrightarrow \mathbb{A}$ *be the bijective function, which maps each point on the unit sphere to Polar Coordinate angles via* $A(s) = \left(P_2^{-1}(s), .., P_N^{-1}(s)\right)^{\intercal}$. *Let* $\mathbb{E} \subseteq \mathbb{S}$. *The probability measure of* $\boldsymbol{s}$ *is*

$$\mathrm{Pr}_{\boldsymbol{s}}(\mathbb{E}) = \int\limits_{A(\mathbb{E})} cK(\alpha)d\alpha, \, \forall \mathbb{E} \in \mathcal{S} \tag{3.59}$$

*characterised by a constant* $c$.

**Corollary 3.18** *The constant* $c = \left( \int\limits_{\mathbb{A}} K(\alpha)d\alpha \right)^{-1}$. *This constant can be calculated from:* $1 = \mathrm{Pr}_{\boldsymbol{s}}(\mathbb{S}) = \int\limits_{A(\mathbb{S})} cK(\alpha)d\alpha = \int\limits_{\mathbb{A}} cK(\alpha)d\alpha$

In the following, the value of constant $c$ is derived, such that an explicit formulation of the DS estimator can be given in Section 3.6.3.

**Lemma 3.2** *Let* $\Gamma : \mathbb{R}_+ \longrightarrow \mathbb{R}_+$ *with* $\Gamma(x) = \int\limits_0^\infty t^{x-1}e^{-t}dt$ *be the Gamma function as defined in Rice (1995).*

$$c = \frac{\Gamma(\frac{\mathrm{N}}{2})}{2(\pi)^{\frac{\mathrm{N}}{2}}} \tag{3.60}$$

**Proof.** The integral over the sample space of a MVN random vector is

$$(2\pi)^{-\mathrm{N}/2} \int\limits_{-\infty}^{\infty} \cdots \int\limits_{-\infty}^{\infty} e^{-\frac{1}{2}v'v} dv = 1 \tag{3.61}$$

[7] $\mathcal{S}$ is the smallest $\sigma$-algebra of all subsets of $\mathbb{S}$

Polar Coordinates Transform applied to expression (3.61) gives

$$
\begin{aligned}
&(2\pi)^{-\frac{\mathrm{N}}{2}} \int\limits_{\mathbb{A}} K(\alpha) \int\limits_{0}^{\infty} r^{\mathrm{N}-1} e^{-\frac{1}{2}r^2} dr \; d\alpha = 1 \\
\Longrightarrow &(2\pi)^{-\frac{\mathrm{N}}{2}} \int\limits_{\mathbb{A}} K(\alpha) \int\limits_{0}^{\infty} w^{\mathrm{N}-1} e^{-w^2} 2^{\frac{1}{2}\mathrm{N}-\frac{1}{2}} 2^{\frac{1}{2}} dw \; d\alpha = 1 \text{ (i.e. transformation: } r = \sqrt{2}w) \\
\Longrightarrow &(2\pi)^{-\frac{\mathrm{N}}{2}} \int\limits_{\mathbb{A}} K(\alpha) \frac{1}{2} \Gamma\left(\frac{\mathrm{N}}{2}\right) 2^{\frac{\mathrm{N}}{2}} \; d\alpha = 1 \\
\Longrightarrow &\left( \int\limits_{\mathbb{A}} K(\alpha) d\alpha \right)^{-1} = \frac{\Gamma(\frac{\mathrm{N}}{2})}{2(\pi)^{\frac{\mathrm{N}}{2}}} \qquad (3.62)
\end{aligned}
$$

■

Notice that $c$ is the inverse of the surface of the unit hyper-sphere.

### 3.6.3 The Directional Sampling (DS) estimator

The theory developed in the previous sections will be used to develop the so-called *Directional Sampling* (DS) estimation method, which is a promising alternative to the well known Monte Carlo estimation method, in the case random vector $\boldsymbol{v}$ has a spherical symmetric distribution.

In Section 3.2 the Happy set Indicator function has been defined in (2.5). Polar Coordinates Transformation of expression (3.55) gives

$$
\begin{aligned}
\Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} &= \int\limits_{\mathbb{H}(x)} g\left(v^{\intercal} v\right) dv = \int\limits_{\mathbb{R}} I(v, x) g\left(v^{\intercal} v\right) dv \\
&= \int\limits_{\mathbb{A}} K(\alpha) \int\limits_{0}^{\infty} I\left(P(r, \alpha), x\right) r^{\;\mathrm{N}-1} g(r^2) dr d\alpha \qquad (3.63)
\end{aligned}
$$

The DS estimation procedure is based on the following theorem

**Theorem 3.5** *Let* $\boldsymbol{s}$ *be a random direction*[8] *and* $c = \frac{2(\pi)^{\frac{\mathrm{N}}{2}}}{\Gamma(\frac{\mathrm{N}}{2})}$

$$
E_{\boldsymbol{s}}\left[ \frac{1}{c} \int\limits_{0}^{\infty} I(r\boldsymbol{s}, x) r^{\mathrm{N}-1} g(r^2) dr \right] = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} \qquad (3.64)
$$

[8] A practical approach for obtaining random direction sample $s$ is the following. Let $v \in \mathbb{R}^{\mathrm{N}}$ be a sample of a MVN random vector. Projecting $v$ on the unit sphere gives random direction sample: $s = \frac{v}{\sqrt{v^{\intercal} v}}$.

**Proof.** Polar Coordinates Transformation of the explicit form of the Expected value gives

$$\begin{aligned}
E_{\boldsymbol{s}}\left[\frac{1}{c}\int\limits_{0}^{\infty} I(r\boldsymbol{s},x)r^{\,\mathrm{N}-1}g(r^2)dr\right] &= \int\limits_{\mathbb{S}}\frac{1}{c}\left(\int\limits_{0}^{\infty} I(rs,x)r^{\,\mathrm{N}-1}g(r^2)dr\right)cds \\
&= \int\limits_{\mathbb{A}} K(\alpha)\int\limits_{0}^{\infty} I\left(r1^{\mathrm{N}-1}P(1,\alpha),x\right)r^{\,\mathrm{N}-1}g(r^2)drd\alpha \\
&= \int\limits_{\mathbb{A}} K(\alpha)\int\limits_{0}^{\infty} I\left(P(r,\alpha),x\right)r^{\,\mathrm{N}-1}g(r^2)drd\alpha \\
&= \Pr\left\{\boldsymbol{v}\in\mathbb{H}(x)\right\}. \qquad (3.65)
\end{aligned}$$

■

**Definition 3.13** *Let $s\in\mathbb{S}$ be a given realisation on the unit hyper-sphere of random direction $\boldsymbol{s}$ and $c=\frac{2(\pi)^{\frac{\mathrm{N}}{2}}}{\Gamma(\frac{\mathrm{N}}{2})}$. The DS estimate function $\overset{ds}{r}:\mathbb{S}\times\mathbb{R}^{\mathrm{I}}\longrightarrow[0,1]$ is defined as*

$$\overset{ds}{r}(s,x)=\frac{1}{c}\int\limits_{0}^{\infty} I(rs,x)r^{\mathrm{N}-1}g(r^2)dr \qquad (3.66)$$

**Corollary 3.19** *Let $\boldsymbol{s}^{[1]},..,\boldsymbol{s}^{[m]}$ be $\mathrm{M}$ independent copies of random direction $\boldsymbol{s}$. From Definition 3.13 and Theorem 3.5 follows directly that the DS-estimator*

$$\overset{ds}{\boldsymbol{R}}(x)=\frac{1}{\mathrm{M}}\sum_{m=1}^{\mathrm{M}}\overset{ds}{r}(\boldsymbol{s}^{[m]},x) \qquad (3.67)$$

*is an unbiased estimator of $R(x)$*

**Corollary 3.20** *Following the definitions of Rice (1995), the standard error[9] of the DS estimator is*

$$se\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)=\sqrt{\frac{1}{\mathrm{M}}\left(E\left[\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)^2\right]-\left(E\left[\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right]\right)^2\right)} \qquad (3.68)$$

[9] Deák (2000) reduced the standard error of the estimator by using antithetic random direction samples and in later research by using orthonormalised random direction samples. It can be shown that using orthonormalisated random directions results in a Directional Sampling estimator with a lower standard error. These topics are not further discussed here, to keep notation as simple as possible.

Consider the multivariate standard normal PDF $f(v) = \frac{1}{(2\pi)^{\mathrm{N}/2}} e^{-\frac{1}{2}v^{\intercal}v} = g(v^{\intercal}v)$, with $g(w) = \frac{1}{(2\pi)^{\mathrm{N}/2}} e^{-\frac{1}{2}w}$. Let M=1 and $s$ be a sample of the random direction $\boldsymbol{s}$. Substitution in (3.67) gives the estimate function introduced by Deák (1986):

$$\begin{aligned}
\overset{\mathrm{ds}}{R}(x) &= \frac{(2\pi)^{\frac{\mathrm{N}}{2}}}{2^{\frac{\mathrm{N}}{2}-1}\Gamma(\frac{\mathrm{N}}{2})} \int_0^\infty I(rs,x) r^{\,\mathrm{N}-1} \frac{1}{(2\pi)^{\mathrm{N}/2}} e^{-\frac{1}{2}r^2} dr \\
&= \int_0^\infty I(rs,x) \frac{1}{2^{\frac{\mathrm{N}}{2}-1}\Gamma(\frac{\mathrm{N}}{2})} r^{\,\mathrm{N}-1} e^{-\frac{1}{2}r^2} dr \\
&= \int_0^\infty I(rs,x) \underbrace{\frac{1}{2^{\frac{\mathrm{N}}{2}}\Gamma(\frac{\mathrm{N}}{2})} y^{\,\mathrm{N}/2-1} e^{-\frac{1}{2}y}}_{\text{PDF of chi-square distribution}} dy
\end{aligned} \tag{3.69}$$

where the last step follows from substituting $r = \sqrt{y}$ and $dr = \frac{1}{2}y^{-\frac{1}{2}}dy$.

### 3.6.4 Comparison of DS and MC estimation method

We will compare the Directional Sampling (DS) estimation method and the Monte Carlo (MC) estimation method, with respect to their accuracy, expressed in terms of standard error (se).

**Theorem 3.6** *Let $\overset{ds}{\boldsymbol{R}}(x)$ be the DS estimator* (3.67) *and $\overset{mc}{\boldsymbol{R}}(x)$ be the MC estimator* (3.1) *and se be its standard error according to* (3.68).

$$se\left(\overset{ds}{\boldsymbol{R}}(x)\right) \leq se\left(\overset{mc}{\boldsymbol{R}}(x)\right) \tag{3.70}$$

**Proof.**
It is sufficient to prove the equation for M = 1, since the standard error of both estimators decreases equally proportional with $\sqrt{\mathrm{M}}$. As $0 \leq \left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)^2 \leq \overset{\mathrm{ds}}{\boldsymbol{R}}(x) \leq 1$ and $\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right)^2 = \overset{\mathrm{mc}}{\boldsymbol{R}}(x)$, it holds that

$$\begin{aligned}
var\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right) &= E\left[\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)^2\right] - \left(E\left[\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right]\right)^2 \\
&\leq E\left[\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)\right] - \left(E\left[\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right]\right)^2 \\
&= E\left[\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right)^2\right] - \left(E\left[\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right]\right)^2 = var\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right)
\end{aligned}$$

For general M, from independency and unbiasedness of both estimators, it follows now

$$se\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right) = \left[VAR(\overset{\mathrm{ds}}{\boldsymbol{R}}(x))\right]^{1/2} \leq \left[VAR(\overset{\mathrm{mc}}{\boldsymbol{R}}(x))\right]^{1/2} = se\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right) \tag{3.71}$$

■

In the following, for a given point $s \in \mathbb{S}$ on the unit hyper-sphere, the set $\{rs | r \in \mathbb{R}_+\}$ is called a *ray*. This set is called a ray, because $v \in \{rs | r \in \mathbb{R}_+\}$ are points on a line including the origin of $\mathbb{R}^{\mathrm{N}}$. The standard error of the MC estimator and the standard error of the DS estimator can be equal. This is the case if the Happy set has the following characteristic.

**Definition 3.14** *Let* $\mathbb{Q}(x) = \left\{ s \in \mathbb{S} \middle| \overset{ds}{r}(s, x) \in \{0, 1\} \right\}$. *The Happy set* $\mathbb{H}(x)$ *is called to have a* DS-Radial-Shape *if* $\mathbb{Q}(x)$ *is dense in* $\mathbb{S}$, *i.e. any ray corresponding* $s \in \mathbb{Q}(x)$ *is either completely inside or completely outside the Happy set .*

If the Happy set has a DS-Radial-Shape, then all realisations of the DS estimator are *almost surely* either 0 or 1. Notice that the range of the MC estimate function is $\{0, 1\}$ if $M = 1$. This similarity leads to the following theorem.

**Theorem 3.7** *For the Happy set* $\mathbb{H}(x)$ *it holds that*

$$se\left(\overset{ds}{\boldsymbol{R}}(x)\right) = se\left(\overset{mc}{\boldsymbol{R}}(x)\right)$$

*if and only if* $\mathbb{H}(x)$ *has a DS-Radial-Shape.*

**Proof.**

In the proof of Theorem 3.6, the equality

$$E\left[\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)^2\right] = E\left[\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right)^2\right]$$

is equivalent to

$$E\left[\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)^2\right] = E\left[\left(\overset{\mathrm{ds}}{\boldsymbol{R}}(x)\right)\right]$$

which holds if and only if any realisation of the estimator $\overset{\mathrm{ds}}{\boldsymbol{R}}(x)$, with M= 1, is almost surely either 0 or 1, i.e. if and only if

$$\Pr\{\boldsymbol{s} \in \mathbb{Q}(x)\} = 1$$

■

Theorem 3.6 and 3.7 lead to

**Corollary 3.21**
*The DS estimator is strictly more accurate than the MC estimator, i.e.*

$$se\left(\overset{ds}{\boldsymbol{R}}(x)\right) < se\left(\overset{mc}{\boldsymbol{R}}(x)\right),$$

*if and only if* $\mathbb{H}(x)$ *does not have a DS-Radial Shape.*

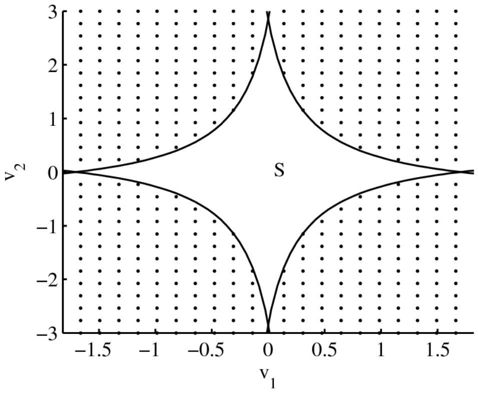


Figure 3.15: The set S is a Ray-Convex set which is not convex.

Let us focus on the computation of the DS estimate.

**Definition 3.15 (Ray-Convexity)** *Let a half-line that starts in the origin be called a* ray. *Let $P$ be the Polar Coordinates Transform and vector $d = P(1, s)$ with $s \in \mathbb{S}$ be the support vector of the ray $\{dr|r \in \mathbb{R}_+\}$. The measurable set $\mathbb{H}(x)$ is called* Ray-Convex *if* $\forall r_1, r_2 \geq 0$,
$s \in \mathbb{S}$, $r_1 s \in \mathbb{H}(x)$ *and* $r_2 s \in \mathbb{H}(x) \Longrightarrow (\alpha r_1 + (1-\alpha) r_2)s \in \mathbb{H}(x)$ *for* $0 \leq \alpha \leq 1$.

Notice that a convex set is also Ray-Convex, but the converse is not true. Figure 3.15 illustrates a Ray-convex set, which is not convex. Figure 3.15 does not show a ray, but this can be depicted as a half-line that starts in the origin.

**Definition 3.16 (Ray intersection distances)** *For a given $s \in \mathbb{S}$, let*

$$a(s) = \inf\{r \in \mathbb{R}_+ | rs \in \mathbb{H}(x)\}$$
$$b(s) = \sup\{r \in \mathbb{R}_+ | rs \in \mathbb{H}(x)\}$$

*Remark that $b(s)$ can be $\infty$.*

Let Happy set $\mathbb{H}(x)$ be Ray-Convex, $s \in \mathbb{S}$ be a given realisation on the unit hyper-sphere of random direction $\boldsymbol{s}$ and $c = \frac{2(\pi)^{\frac{N}{2}}}{\Gamma(\frac{N}{2})}$. The DS estimate function of Definition 3.13 can be determined via:

$$\begin{aligned} \overset{\mathrm{ds}}{r}(s,x) &= c \int_0^\infty I(r\boldsymbol{s}^{[m]}, x) r^{\ \mathrm{N}-1} g(r^2) dr \\ &= H(b(s)) - H(a(s)) \end{aligned} \tag{3.72}$$

where

$$H(x) = c \int_0^x r^{\mathrm{N}-1} g(r^2) dr \tag{3.73}$$

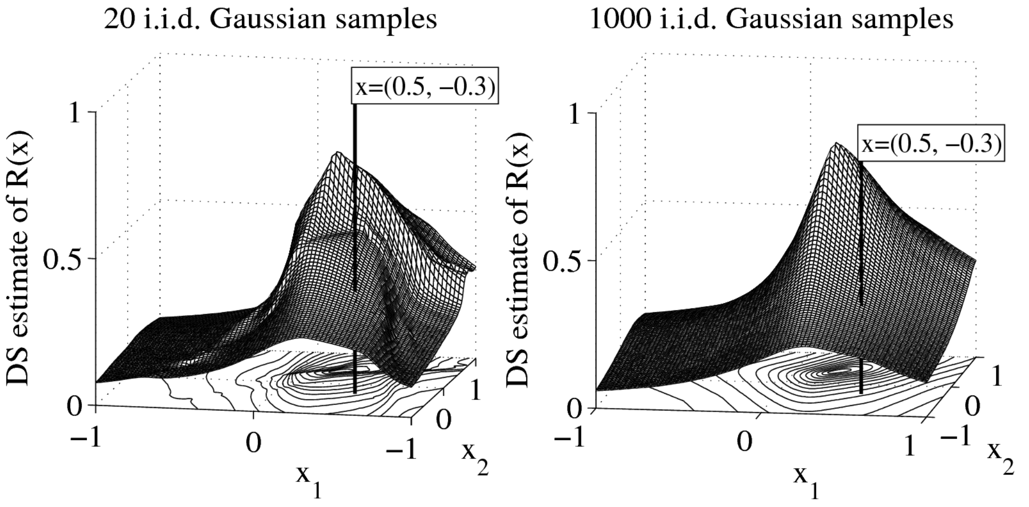


Figure 3.16: Illustration of DS estimation for 20 and 1000 samples

where (3.73) is a convenient form for numerical evaluation. For a measurable set $\mathbb{H}(x)$ which is a finite union $\bigcup_{k=1}^{\mathrm{K}} \mathbb{H}^{[k]}(x)$ of K disjunct Ray-Convex sets $\mathbb{H}^{[k]}(x)$, the probability estimator of $\mathbb{H}(x)$ is the sum of the DS-estimators of the individual sets.

**Example 3.6** *Consider the Robustness function $R(x)$ and random vector $\boldsymbol{v}$ as defined in the "Gaussian" situation of Example 3.1. The mesh surface of the DS estimate function, based on 20 samples and 1000 samples are shown in Figure 3.16.*

*Note that the mesh surfaces in Figure 3.16 look more smooth than the mesh surfaces of respectively the MC method in Figure 3.1 and the SMC method in Figure 3.9 . The standard error of the DS estimate in Table 3.4, relative to the standard error of the MC method in Table 3.1 and SMC method in Table 3.2, shows that the DS method is more accurate, given the same number of samples.*

*We compare the optimisation of the DS estimate function to the optimisation of the MC estimate function of Example 3.3: The starting point is $x^{[0]} = (0.5, -0.3)$; FD step-sizes are $\Delta \in \left\{ \left(\frac{1}{2}\right)^{10}, \left(\frac{1}{2}\right)^{9}, ..., \left(\frac{1}{2}\right)^{1}, \left(\frac{1}{2}\right)^{0} \right\}$; All other parameters are kept at the Matlab default values. Table 3.4 gives the values of $\overset{ds}{R}{}^*$ to which the SQP algorithm converged.*

*From Table 3.4 follows that DS yields better results than MC. However DS, N-1MC and SMC appear to be competitive for this example. The best result given 1000 samples, was found with $\Delta = \left(\frac{1}{2}\right)^{9}$. The corresponding optimisation results are shown in Figure 3.17. The line pieces, from the origin in the direction of the samples, in Figure 3.17, illustrate the intervals between $a(s^{[m]})$ and $b(s^{[m]})$, which are input for the DS estimate function (3.72) .*

Table 3.4: SQP Optimisation results of DS estimate function, for a given $\Delta$, sample size and Normal distribution

| | M=20 | | M=1000 | |
|---|---|---|---|---|
| $\Delta$ | $\overset{\text{ds}}{R}^*$ | $\tilde{R}\,(x^*)$ | $\overset{\text{ds}}{R}^*$ | $\tilde{R}\,(x^*)$ |
| Robustness estimate at starting point $x^{[0]} = (0.5, -0.3)$: | | | | |
| | 0.346 | 0.293 | 0.297 | 0.293 |
| $\left(\frac{1}{2}\right)^{10}$ | 0.753 | 0.775 | 0.777 | 0.777 |
| $\left(\frac{1}{2}\right)^{9}$ | 0.757 | 0.780 | **0.782** | **0.782** |
| $\left(\frac{1}{2}\right)^{8}$ | 0.757 | 0.780 | 0.776 | 0.776 |
| $\left(\frac{1}{2}\right)^{7}$ | 0.752 | 0.774 | 0.780 | 0.780 |
| $\left(\frac{1}{2}\right)^{6}$ | **0.758** | **0.780** | 0.767 | 0.767 |
| $\left(\frac{1}{2}\right)^{5}$ | 0.750 | 0.775 | 0.771 | 0.772 |
| $\left(\frac{1}{2}\right)^{4}$ | 0.632 | 0.674 | 0.764 | 0.764 |
| $\left(\frac{1}{2}\right)^{3}$ | 0.620 | 0.639 | 0.778 | 0.778 |
| $\left(\frac{1}{2}\right)^{2}$ | 0.512 | 0.480 | 0.573 | 0.573 |
| $\left(\frac{1}{2}\right)^{1}$ | 0.426 | 0.412 | 0.443 | 0.444 |
| $\left(\frac{1}{2}\right)^{0}$ | 0.346 | 0.293 | 0.301 | 0.299 |
| Average computation time and average estimated standard error per function evaluation are: | | | | |
| time (sec.) | 0.0058 | 141.864 | 0.2699 | 135.228 |
| $\hat{se}$ | 0.0499 | 0.0005 | 0.0069 | 0.0005 |

Comments on Table 3.4:

1. $\overset{\text{ds}}{R}^*$ is the Robustness estimate based on M samples, of point $x^*$ where SQP converged to.
2. $\tilde{R}(x^*)$ is an MC estimate of $R(x^*)$, with $\hat{se}\left(\tilde{\boldsymbol{R}}\,(x^*)\right) < 0.0005$ (See Appendix A.3).

## 3.7 Exponential Simplex estimation method

The Exponential Simplex estimation method has been designed for Robustness Programming, where elements of $\boldsymbol{v}$ are i.i.d. Exponentially distributed (see Olieman and Van Putten, 2006). The Exponential distribution plays a central role in Reliability Theory and Reliability Engineering (Terje and Uwe, 1999; Rausand and Høyland, 2004). The Exponential distribution is typically used to model the life-time, or the time until the next failure of a (sub)system. The reliability of a system can be analysed, by decomposing a system into subsystems. Typically, the failure of each subsystem is modelled by a separate Exponential distribution. Hence, the failure of the complete system is modelled by a multivariate Exponential distribution. This way, the Reliability of a system is equivalent to the Robustness of the system and can be quantified as the probability of failure before a certain moment in time. The failure of the complete system is characterised by a subset of the sample space of such multivariate Exponential distribution and the Happy set is the complement of this set. A reliability optimisation problem, involving Exponentially distributed uncertain factors, is for example discussed by Azaron et al. (2007)

Queuing Theory studies queuing systems or queuing networks and is another area in which the Exponential distribution plays a central role. The inter-arrival times and service times are often assumed to follow an Exponential distribution (Law and Kelton, 2000). Such queuing systems are called M/M/$s$ queues, where the symbol M refers to

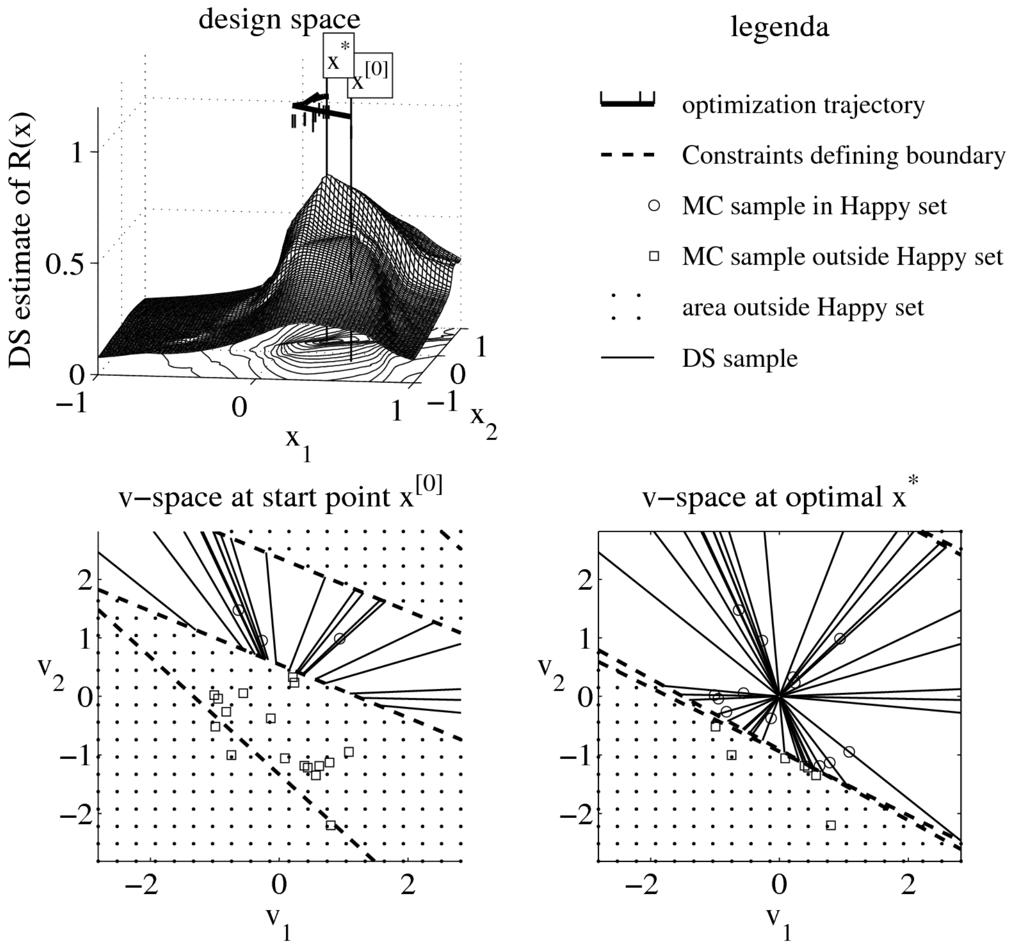


Figure 3.17: SQP DS estimate function optimisation results; Gaussian situation; M=20; $\Delta = \left(\frac{1}{2}\right)^6$

the Markovian, i.e. memory-less, property of the Exponentially distributed arrival times and service times and $s$ is the number of servers. An interesting Robustness performance measure of a queuing system, is the probability that a customer is serviced within an acceptable time. Inter-linked queuing systems like traffic networks (such as telecommunication networks, railroads and motor-ways) are too complex for analytical analysis. Such systems are typically analysed via simulation instead (Law and Kelton, 2000).

In principle, Monte Carlo sampling can be used in both the context of Reliability Theory and Queuing Theory, to estimate respectively the Robustness of the lifetime of a system or the Robustness of timely servicing in a queuing system.

However, the arguments given in Section 3.3 suggest that the MC method is not convenient in the context of optimising Robustness. Therefore, if the objective is to design a system in such a way that reliability is optimised, or when the objective is to design a queuing system with optimal Robustness with respect to the timely servicing, then it is relevant to have an alternative for the MC estimation approach with better optimisation characteristics.

In this section, the Exponential Simplex (ES) estimation method is introduced as a better alternative for the MC estimation method, which can be used to estimate the probability mass of a multivariate Exponential distribution over the Happy set. The Exponential Simplex estimation method, for estimating the probability mass of a measurable set, is discussed in detail by Olieman and Van Putten (2006).

In general, there are no analytical solutions for the computation of the probability mass of a multivariate exponential distribution over an arbitrary set (see for instance Anderson, 1984; Narayan, 1996). The ES method is a new technique for estimating such a probability. The inspiration for the concept originates from the works of István Deák (1986), Alan Genz (1993) and Paul Somerville (1998) who studied integration of multivariate normal distributions and practical experience with probability estimation methods in a game theoretic context (illustrated in Olieman and Hendrix, 2006; Dellink et al., 2007).

The following notation is used. The random vector $\boldsymbol{v}$ with non-negative realisations $v = [v_1, v_2, ..., v_{\mathrm{N}}] \in \mathbb{V} = \mathbb{R}_+^{\mathrm{N}}$, has independent Exponentially distributed elements $v_i \sim Exp(1)$, such that $E(\boldsymbol{v}_i) = VAR(\boldsymbol{v}_i) = 1$. The Probability Density Function (PDF) of $\boldsymbol{v}$ is the function $f : \mathbb{R}^{\mathrm{N}} \longrightarrow \mathbb{R}_+$ with

$$f(v) = \begin{cases} \prod_{i=1}^{\mathrm{N}} \exp\left(-v_i\right) = \exp\left(-\sum_{i=1}^{\mathrm{N}} v_i\right) & \text{for } v \in \mathbb{R}_+^{\mathrm{N}} \\ 0 & \text{elsewhere} \end{cases} \tag{3.74}$$

The purpose of the estimation technique is to estimate (2.4) . The following feature will be exploited by the ES estimation method.

**Definition 3.17** *An iso-probability density contour of function $f$, is a* positive orthant simplex *defined by $\sum_{i=1}^{\mathrm{N}} v_i = r$ with $r \geq 0$ and $v = [v_1, v_2, .., v_{\mathrm{N}}] \in \mathbb{R}_+^{\mathrm{N}}$.*

This means that for a given $r$, $f(v)$ is constant for any $v$ with $\sum_{i=1}^{\mathrm{N}} v_i = r$.

### 3.7.1 Simplex Coordinates Transform

The Robustness expression in (2.4) does not have an analytical solution in general. As an analogue of the polar coordinates transformation (discussed in Section 3.6.1), in which the unit circle plays a central role, a "Simplex Coordinates Transformation" is defined, in which the unit simplex takes the role of the unit circle in the positive orthant. The transformation induces an equivalent expression for (2.4) , which holds useful properties for estimation.

**Definition 3.18** *The Simplex Coordinates Transformation (SCT) is defined by the fol-*

*lowing mapping*

$$T : \mathbb{Q} \longrightarrow \mathbb{R}_+^{\mathrm{N}} \text{ with} \tag{3.75}$$

$$T(r, s_1, s_2, ..., s_{\mathrm{N}-1}) = \left( rs_1, rs_2, ..., rs_{\mathrm{N}-1}, r(1 - \sum_{i=1}^{\mathrm{N}-1} s_i) \right) \tag{3.76}$$

*where* $\mathbb{Q} = \mathbb{R}_+ \times \mathbb{S}^{\mathrm{N}-1}$ *and* $\mathbb{S}^{\mathrm{N}-1} = \left\{ s \in \mathbb{R}_+^{\mathrm{N}-1} \middle| \sum_{i=1}^{\mathrm{N}-1} s_i \leq 1 \right\}$.

Note that for $r = 1$, coordinates from the lower dimensional space $\mathbb{S}^{\mathrm{N}-1}$ are mapped on unit simplex coordinates in $\mathbb{R}^{\mathrm{N}}$. The variable $r \in \mathbb{R}_+$ can be interpreted as a multiplying factor, similar to the radius variable in polar coordinates transformation. Notice that $T(1, s)$ with $s \in \mathbb{S}^{\mathrm{N}-1}$ are points on the unit simplex and for the general case $T(r, s) = rT(1, s)$ with $r \in \mathbb{R}_+$.

Transformation $T$ has the following properties, which are essential for giving a proper expression for (2.4) in Simplex Coordinates (see for instance Marsden and Tromba, 1996):

- $T$ is a.e.[10] injective: $T(q_1) \neq T(q_2)$ if $q_1 \neq q_2$, for $q_1, q_2 \in \mathbb{Q}$
- $T$ is surjective ("onto"): $\bigcup_{q \in \mathbb{Q}} T(q) = \mathbb{R}_+^{\mathrm{N}}$
- The absolute value of the Jacobian determinant of $T$ gives the change of integration measure implied by the transformation, where the Jacobian determinant $J : \mathbb{Q} \longrightarrow \mathbb{R}$ is

$$J(q) = \det \frac{\partial T(q)}{\partial q} = \det \begin{bmatrix} s_1 & r & 0 & 0 & \cdots & 0 \\ s_2 & 0 & r & 0 & \cdots & 0 \\ \vdots & \vdots & & \ddots & & \vdots \\ s_{\mathrm{N}-1} & 0 & 0 & 0 & \cdots & r \\ 1 - s_1 - s_2 - ... - s_{\mathrm{N}-1} & -r & -r & -r & \cdots & -r \end{bmatrix}$$
$$= (-1)^{\mathrm{N}} r^{\mathrm{N}-1} \tag{3.77}$$

The latter step can be obtained by summing up the first $\mathrm{N} - 1$ rows to the last row and by expanding the resulting determinant along its last row. Notice that $J(q)$ has constant sign with absolute value equal to $r^{\mathrm{N}-1}$.

These properties imply that (2.4) can be rewritten as

$$\Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} = \int_{\mathbb{S}^{\mathrm{N}-1}} \int_0^\infty I^{[SC]}(x, r, s) \exp(-r) |J(r, s)| dr ds$$
$$= \int_{\mathbb{S}^{\mathrm{N}-1}} \int_0^\infty I^{[SC]}(x, r, s) \exp(-r) r^{\mathrm{N}-1} dr ds \tag{3.78}$$

[10] Transformation $T$ is almost everywhere (a.e.) injective, i.e. the injective property of $T$ holds apart from a set of measure zero (the origin, in this case). Observe that the widely used Polar Coordinates transform has the same property.

where $I^{[SC]} : \mathbb{X} \times \mathbb{Q} \longrightarrow \{0,1\}$ is the indicator function of $\mathbb{H}(x)$ expressed in Simplex Coordinates:

$$I(x, T(r,s)) = I^{[SC]}(x,r,s) = \left\{ \begin{array}{ll} 1 & \text{if } T(r,s) \in \mathbb{H}(x) \\ 0 & \text{elsewhere} \end{array} \right. \tag{3.79}$$

and factor $\exp(-r)$ is obtained via

$$f\left(T(r,s)\right) = \exp\left( -\sum_{i=1}^{\mathrm{N}-1} r s_i \right) \exp\left( -r(1 - \sum_{i=1}^{\mathrm{N}-1} s_i) \right) = \exp(-r) \tag{3.80}$$

for $r \in \mathbb{R}_+$ and $s \in \mathbb{S}^{\mathrm{N}-1}$.

### 3.7.2 The Exponential Simplex (ES) Estimator

The theoretical observations and transformation will be used to develop the so-called *Exponential Simplex* (ES) estimation method. This is a promising alternative to the well known Monte Carlo estimation method, in the case of exponentially distributed random variables. The estimation procedure is based on the following theorem

**Theorem 3.8** *Let $\boldsymbol{s}$ be a random vector*[11] *uniformly distributed on the unit simplex* $\mathbb{S}^{\mathrm{N}-1}$.

$$E_{\boldsymbol{s}} \left[ \frac{1}{(\mathrm{N}-1)!} \int_0^\infty I^{[SC]}(x,r,\boldsymbol{s}) \exp(-r) r^{\mathrm{N}-1} dr \right] = \Pr\left\{ \boldsymbol{v} \in \mathbb{H}(x) \right\} \tag{3.81}$$

**Proof.** From standard calculus it follows that the *volume* of the sample space $\mathbb{S}^{\mathrm{N}-1}$ of $\boldsymbol{s}$ is $\int\limits_{\mathbb{S}^{\mathrm{N}-1}} 1 ds = \frac{1}{(\mathrm{N}-1)!}$. Therefore the PDF of $\boldsymbol{s}$ is defined by a constant function

$$g(s) = \left\{ \begin{array}{cc} (\mathrm{N}-1)! & \text{for } s \in \mathbb{S}^{\mathrm{N}-1} \\ 0 & \textit{elsewhere} \end{array} \right.$$

The left-hand side of expression (3.81) leads to integral (3.78) via

$$\begin{aligned} & E_{\boldsymbol{s}} \left[ \frac{1}{(\mathrm{N}-1)!} \int_0^\infty I^{[SC]}(x,r,\boldsymbol{s}) \exp(-r) r^{\mathrm{N}-1} dr \right] \\ = & \int\limits_{\mathbb{S}^{\mathrm{N}-1}} g(s) \int_0^\infty \frac{1}{(\mathrm{N}-1)!} I^{[SC]}(x,r,s) \exp(-r) r^{\mathrm{N}-1} dr ds \\ = & \int\limits_{\mathbb{S}^{\mathrm{N}-1}} \frac{(\mathrm{N}-1)!}{(\mathrm{N}-1)!} \int_0^\infty I^{[SC]}(x,r,s) \exp(-r) r^{\mathrm{N}-1} dr ds \end{aligned} \tag{3.82}$$

which is equivalent to probability $\Pr\left\{ \boldsymbol{v} \in \mathbb{H}(x) \right\}$ ■

[11] For the application of such a method, it is relevant to mention that methods for generating random points in a polytope (such as $\mathbb{S}^{N-1}$) are discussed in detail in Devroye (1986).

**Definition 3.19** *Let* $s \in \mathbb{S}^{\text{N-1}}$ *be a given realisation of* $\boldsymbol{s}$. *The ES estimate function* $\overset{es}{r} : \mathbb{S}^{\text{N-1}} \times \mathbb{R}^{I} \longrightarrow [0,1]$ *is defined as*

$$\overset{es}{r}(s,x) = \frac{1}{(\mathrm{N}-1)!} \int\limits_{0}^{\infty} I^{[SC]}(x,r,s) \exp(-r) r^{\mathrm{N}-1} dr \tag{3.83}$$

**Corollary 3.22** *Let* $\boldsymbol{s}^{[1]}, .., \boldsymbol{s}^{[m]}$ *be* M *independent copies of the random vector* $\boldsymbol{s}$. *From Definition 3.19 and Theorem 3.8 follows that the estimator*

$$\overset{es}{\boldsymbol{R}}(x) = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} \overset{es}{r}\left(\boldsymbol{s}^{[m]}, x\right) \tag{3.84}$$

*is an unbiased estimator of* $R(x)$

**Corollary 3.23** *Following the definitions of Rice (1995), the standard error of the estimator is*

$$se\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right) = \sqrt{\frac{1}{\mathrm{M}}\left(E\left[\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right)^2\right] - \left(E\left[\overset{\text{es}}{\boldsymbol{R}}(x)\right]\right)^2\right)} \tag{3.85}$$

Let us focus on the computation of the ES estimate function. For the discussion it is convenient to define the following.

**Definition 3.20** *For fixed* $s \in \mathbb{S}^{\mathrm{N}-1}$, *let*

$$\begin{aligned} a(s) &= \inf\left\{r \in \mathbb{R}_{+} | T(r,s) \in \mathbb{H}(x)\right\} \\ b(s) &= \sup\left\{r \in \mathbb{R}_{+} | T(r,s) \in \mathbb{H}(x)\right\} \end{aligned}$$

Remark that $b(s)$ can take the value $\infty$. For a given $s \in \mathbb{S}^{\mathrm{N}-1}$, the ES estimate can be computed analytically with

$$\overset{\text{es}}{r}(s,x) = \frac{1}{(\mathrm{N}-1)!} \int\limits_{a(s)}^{b(s)} \exp(-r) r^{\mathrm{N}-1} dr \tag{3.86}$$

if

$$a(s) < t < b(s) \Longrightarrow t \in \left\{r \in \mathbb{R}_{+} | T(r,s) \in \mathbb{H}(x)\right\} \tag{3.87}$$

To make a distinction between the ES estimate function $\overset{\text{es}}{r}(s,x)$ of Definition 3.19 and the special case (3.86) if condition (3.87) applies, we define the following

**Definition 3.21** *The* CDF $H : \mathbb{R}_{+} \longrightarrow [0,1]$ *is defined as*

$$H(x) = \frac{1}{(\mathrm{N}-1)!} \int\limits_{0}^{x} \exp(-r) r^{\mathrm{N}-1} dr \tag{3.88}$$

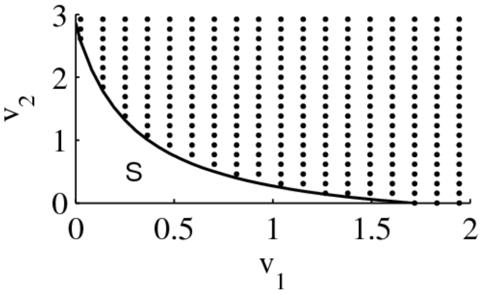


Figure 3.18: The set S is an ES-Ray-Convex set which is not convex.

It can be shown that an analytical expression exists for $H(x)$.

**Corollary 3.24**

$$H(x) = 1 - \exp(-x) \sum_{i=0}^{\mathrm{N}-1} \frac{x^i}{i!} \tag{3.89}$$

If the Happy set has the following property, then (3.87) is true for all $s \in \mathbb{S}^{\mathrm{N}-1}$:

**Definition 3.22 (ES-Ray-Convexity)** *Let a half-line that starts in the origin be called a* ray*. Let $T$ be the Simplex Coordinates Transform and $d = T(1, s)$ with given $s \in \mathbb{S}^{\mathrm{N}-1}$, that defines the ray $\{dr | r \in \mathbb{R}_+\}$. The Happy set $\mathbb{H}(x)$ is called* ES-Ray-Convex *if $\forall s \in \mathbb{S}^{\mathrm{N}-1}$, $T(r_1, s) \in \mathbb{H}(x)$ and $T(r_2, s) \in \mathbb{H}(x)$ with $r_1, r_2 \geq 0 \Longrightarrow T(\alpha r_1 + (1-\alpha) r_2, s) \in \mathbb{H}(x)$ for $0 \leq \alpha \leq 1$.*

This means that a Happy set is called ES-Ray-Convex, if two points $a$ and $b$ on a ray in $\mathbb{R}_+^{\mathrm{N}}$ in the Happy set implies that all points between $a$ and $b$ are in the Happy set. Notice that a convex set is also ES-Ray-Convex, but the converse is not true. Figure 3.18 illustrates a ES-Ray-Convex set which is not convex. Figure 3.18 does not show a ray, but this can be depicted as a half-line in the positive orthant that starts in the origin.

For a ES-Ray-Convex Happy set, the ES estimator and ES estimate function can be based on the analytical expression (3.89).

**Corollary 3.25** *Let the Happy set $\mathbb{H}(x)$ be ES-Ray-Convex, then the ES estimator is*

$$\overset{es}{\boldsymbol{R}}(x) = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} \left\{ H\left(b(\boldsymbol{s}^{[m]})\right) - H\left(a(\boldsymbol{s}^{[m]})\right) \right\} \tag{3.90}$$

**Corollary 3.26** *Let the Happy set $\mathbb{H}(x)$ be ES-Ray-Convex and $s^{[1]}, .., s^{[m]}, .., s^{[M]}$ be* M *independent samples of $\boldsymbol{s}$, then the ES estimate function is*

$$\overset{es}{R}(x) = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} \left\{ H\left(b(s^{[m]})\right) - H\left(a(s^{[m]})\right) \right\} \tag{3.91}$$

*with corresponding standard error estimate*

$$\hat{se}\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right) = \sqrt{\frac{1}{\mathrm{M}-1}\left(\frac{1}{\mathrm{M}}\sum_{m=1}^{\mathrm{M}}\left\{H\left(b(s^{[m]})\right) - H\left(a(s^{[m]})\right\}^2 - \left(\overset{\text{es}}{R}(x)\right)^2\right)} \tag{3.92}$$

For a measurable set $\mathbb{H}(x)$ which is a finite union $\bigcup_{k=1}^{\mathrm{K}}\mathbb{H}^{[k]}(x)$ of K disjunct ES-Ray-Convex sets $\mathbb{H}^{[k]}(x)$, the probability estimator of $\mathbb{H}(x)$ is the sum of the ES-estimators of the individual sets.

### 3.7.3 Comparison of ES and MC estimation methods

We will compare the performance of the Exponential Simplex (ES) estimation method and the Monte Carlo (MC) estimation method, with respect to their accuracy, expressed in terms of standard error (se).

**Theorem 3.9** *Let* $\overset{es}{\boldsymbol{R}}(x)$ *be the ES estimator* (3.84) *and* $\overset{mc}{\boldsymbol{R}}(x)$ *be the MC estimator* (3.1) *and se be its standard error according to* (3.85).

$$se\left(\overset{es}{\boldsymbol{R}}(x)\right) \leq se\left(\overset{mc}{\boldsymbol{R}}(x)\right) \tag{3.93}$$

**Proof.**
It is sufficient to prove the equation for M = 1, since the standard error of both estimators decreases equally proportional with $\sqrt{\mathrm{M}}$. As $0 \leq \left(\overset{\text{es}}{\boldsymbol{R}}(x)\right)^2 \leq \overset{\text{es}}{\boldsymbol{R}}(x) \leq 1$ and $\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)^2 = \overset{\text{mc}}{\boldsymbol{R}}(x)$, it holds that

$$\begin{aligned}
var\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right) &= E\left[\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right)^2\right] - \left(E\left[\overset{\text{es}}{\boldsymbol{R}}(x)\right]\right)^2 \\
&\leq E\left[\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right)\right] - \left(E\left[\overset{\text{es}}{\boldsymbol{R}}(x)\right]\right)^2 \\
&= E\left[\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)^2\right] - \left(E\left[\overset{\text{mc}}{\boldsymbol{R}}(x)\right]\right)^2 = var\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)
\end{aligned}$$

For general M, from independency and unbiasedness of both estimators, it follows now

$$se\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right) = \left[VAR(\overset{\text{es}}{\boldsymbol{R}}(x))\right]^{1/2} \leq \left[VAR(\overset{\text{mc}}{\boldsymbol{R}}(x))\right]^{1/2} = se\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right) \tag{3.94}$$

■

The standard error of the MC estimator and the standard error of the ES estimator can be equal. This is the case if the Happy set has the following characteristic.

**Definition 3.23** *Let* $\overset{es}{r}(s,x)$ *be defined in* (3.83) *and* $\mathbb{Q}(x) = \left\{s \in \mathbb{S}^{N-1} \,\middle|\, \overset{es}{r}(s,x) \in \{0,1\}\right\}$. *The Happy set* $\mathbb{H}(x)$ *has a* ES-Radial-Shape *if* $\mathbb{Q}(x)$ *is dense in* $\mathbb{S}^{N-1}$, *i.e. a ray corresponding to* $s \in \mathbb{Q}(x)$ *is either completely inside or completely outside the Happy set* .

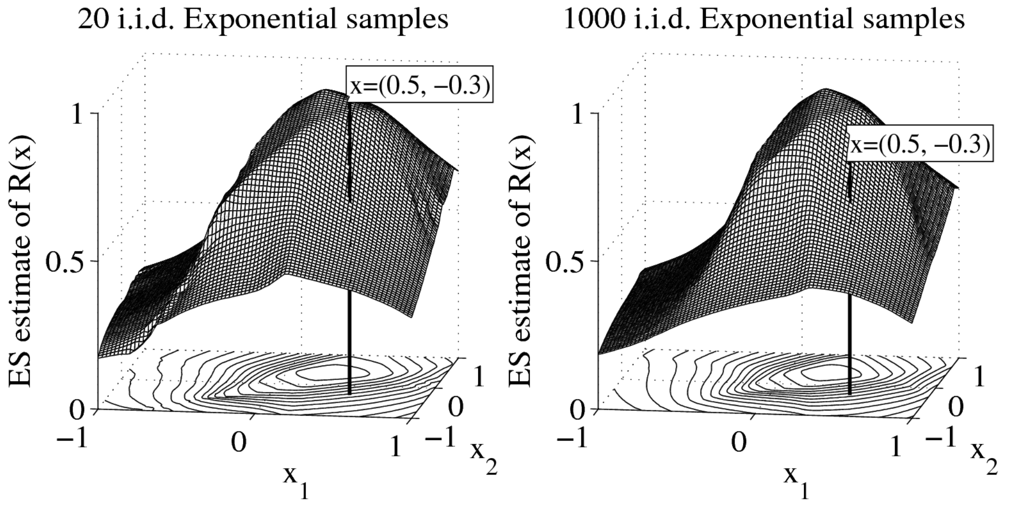


Figure 3.19: Illustration of ES estimation for 20 and 1000 samples

If $M = 1$ and the Happy set has a ES-Radial-Shape, then all realisations of the ES estimator are *almost surely* (a.s.) either 0 or 1. Hence, this range is a.s. identical to the range of the MC estimate function for M=1.

**Theorem 3.10** *For the Happy set* $\mathbb{H}(x)$ *holds that*

$$se\left(\overset{es}{\boldsymbol{R}}(x)\right) = se\left(\overset{mc}{\boldsymbol{R}}(x)\right)$$

*if and only if* $\mathbb{H}(x)$ *has a ES-Radial-Shape.*

**Proof.**

Let M= 1, then $\overset{\text{es}}{\boldsymbol{R}}(x) = \overset{\text{es}}{r}(\boldsymbol{s}, x)$ is implied by (3.84). In the proof of Theorem 3.9, the equality

$$E\left[\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right)^2\right] = E\left[\left(\overset{\text{mc}}{\boldsymbol{R}}(x)\right)^2\right]$$

is equivalent to

$$E\left[\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right)^2\right] = E\left[\left(\overset{\text{es}}{\boldsymbol{R}}(x)\right)\right]$$

which holds if and only if a realisation of the estimator $\overset{\text{es}}{\boldsymbol{R}}(x)$ is a.s. either 0 or 1, i.e. if and only if the set

$$\mathbb{Q}(x) = \left\{s \in \mathbb{S}^{\mathrm{N}-1} \,\middle|\, \overset{\text{es}}{r}(s, x) \in \{0, 1\}\right\}$$

is dense in $\mathbb{S}^{\mathrm{N}-1}$, i.e. $\Pr\{\boldsymbol{s} \in \mathbb{Q}(x)\} = 1$. ■

Theorem 3.9 and 3.10 lead to

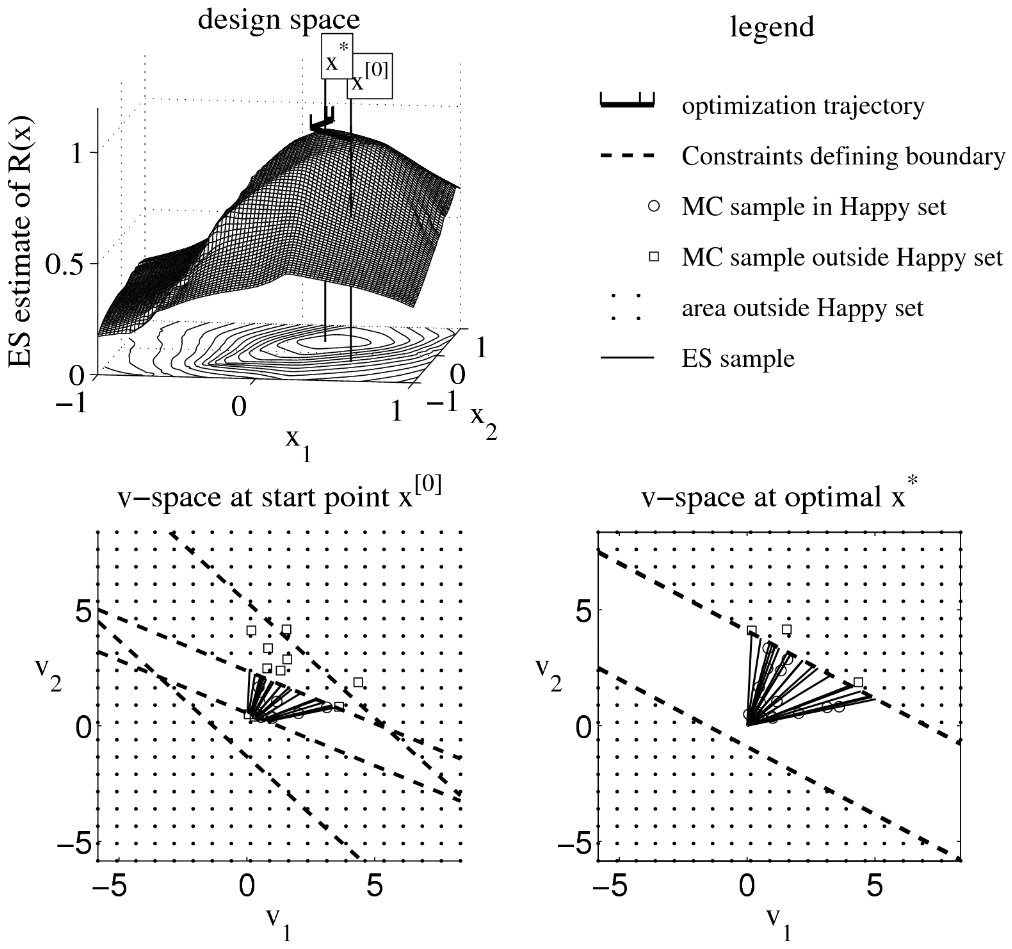


Figure 3.20: SQP ES estimate function optimisation results; Exponential situation; M=20; $\Delta = \left(\frac{1}{2}\right)^{10}$

**Corollary 3.27**
*The ES estimator is strictly more accurate than the MC estimator, i.e.*

$$se\left(\overset{es}{\boldsymbol{R}}(x)\right) < se\left(\overset{mc}{\boldsymbol{R}}(x)\right),$$

*if and only if* $\mathbb{H}(x)$ *does not have a ES-Radial-Shape.*

For a radially shaped set $\mathbb{H}(x)$, the difference between the MC standard error and ES standard error is minimal (i.e. is zero). Conversely, the maximum difference in standard error occurs when $\mathbb{H}(x)$ is a dense set in the positive orthant, bounded by a plane parallel to the unit simplex, such as $u(x, v) = \sum_{n=1}^{\mathrm{N}} v_n \leq$H, H$\geq 0$. In this situation $a(s) = 0$ and $b(s) =$H for all $s \in \mathbb{S}^{\mathrm{N}-1}$ and consequently all realisations of the ES-estimator are exactly $\Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$. In this case the standard error of the ES-estimator is zero whereas the standard error of MC-estimator is $(\frac{1}{\mathrm{M}}p(1-p))^{\frac{1}{2}}$ with maximum $\frac{1}{2\sqrt{M}}$ in case $p = \frac{1}{2}$.

Table 3.5: SQP Optimisation results of ES estimate function, for a given $\Delta$, sample size and Exponential distribution

| | M=20 | | M=1000 | |
|---|---|---|---|---|
| $\Delta$ | $\overset{\text{es}}{\tilde{R}}{}^*$ | $\tilde{R}\left(x^*\right)$ | $\overset{\text{es}}{\tilde{R}}{}^*$ | $\tilde{R}\left(x^*\right)$ |
| Robustness estimate at starting point $x^{[0]} = (0.5, -0.3)$: | | | | |
| | 0.647 | 0.642 | 0.644 | 0.642 |
| $\left(\frac{1}{2}\right)^{10}$ | **0.959** | **0.960** | **0.961** | **0.960** |
| $\left(\frac{1}{2}\right)^{9}$ | 0.959 | 0.960 | 0.961 | 0.960 |
| $\left(\frac{1}{2}\right)^{8}$ | 0.958 | 0.960 | 0.960 | 0.959 |
| $\left(\frac{1}{2}\right)^{7}$ | 0.959 | 0.960 | 0.960 | 0.959 |
| $\left(\frac{1}{2}\right)^{6}$ | 0.957 | 0.958 | 0.959 | 0.958 |
| $\left(\frac{1}{2}\right)^{5}$ | 0.954 | 0.955 | 0.957 | 0.956 |
| $\left(\frac{1}{2}\right)^{4}$ | 0.954 | 0.955 | 0.958 | 0.957 |
| $\left(\frac{1}{2}\right)^{3}$ | 0.943 | 0.945 | 0.956 | 0.955 |
| $\left(\frac{1}{2}\right)^{2}$ | 0.923 | 0.916 | 0.923 | 0.921 |
| $\left(\frac{1}{2}\right)^{1}$ | 0.868 | 0.858 | 0.847 | 0.845 |
| $\left(\frac{1}{2}\right)^{0}$ | 0.712 | 0.642 | 0.645 | 0.644 |
| Average computation time and average estimated standard error per function evaluation are: | | | | |
| time (sec.) | 0.0084 | 48.380 | 0.3778 | 48.144 |
| $\hat{se}$ | 0.0255 | 0.0005 | 0.0042 | 0.0005 |

Comments on Table 3.5:

1. $\overset{\text{es}}{\tilde{R}}{}^*$ is the Robustness estimate based on M samples, of point $x^*$ where SQP converged to.
2. $\tilde{R}(x^*)$ is an MC estimate of $R(x^*)$, with $\hat{se}\left(\tilde{\boldsymbol{R}}\left(x^*\right)\right) < 0.0005$ (See Appendix A.3).

The optimisation of $\overset{es}{\tilde{\boldsymbol{R}}}(x)$ is illustrated with an extension of Example 3.1.

**Example 3.7** *Consider the Robustness function $R(x)$ and random vector $\boldsymbol{v}$ as defined by the "Exponential" situation of Example 3.1. The mesh surface of the ES estimate function, based on 20 samples and 1000 samples are shown in Figure 3.19.*

*Note that the mesh surfaces in Figure 3.19 look more smooth than the mesh surfaces of respectively the MC method in Figure 3.1 and the SMC method in Figure 3.9. The standard error of the ES estimate in Table 3.5, relative to the standard error of the MC method in Table 3.1 and SMC method in Table 3.2, shows that the ES method is almost 4 times as accurate, given the same number of samples.*

*Optimisation of the ES estimate function is compared to the optimisation of the MC estimate function of Example 3.3: The starting point is $x^{[0]} = (0.5, -0.3)$; FD step-sizes are $\Delta \in \left\{\left(\frac{1}{2}\right)^{10}, \left(\frac{1}{2}\right)^{9}, ..., \left(\frac{1}{2}\right)^{1}, \left(\frac{1}{2}\right)^{0}\right\}$; All other parameters are kept at the Matlab default values. Table 3.5 gives the values of $\overset{es}{\tilde{R}}{}^*$ to which the SQP algorithm converged.*

*Table 3.5 shows that ES yields better results than MC. However ES, N-1MC and SMC appear to be competitive for this example. The best result given 20 samples, was found with $\Delta = \left(\frac{1}{2}\right)^{10}$. The corresponding optimisation results are shown in Figure 3.20. The line pieces, from the origin in the direction of the samples, in Figure 3.20, illustrate the intervals between $a(s^{[m]})$ and $b(s^{[m]})$, which are input for the ES estimate function* (3.91).

Apart from being superior to the MC-estimator in the above mentioned statistical sense, another advantage for ES-Ray-Convex sets of the ES-estimator, is that it can be expressed in a very simple analytical form following Corollary 3.25. In case of a finite union of disjunct ES-Ray-Convex sets, the ES-estimator takes the form of the sum of the corresponding analytical expressions. We remark that the class of (finite unions of disjunct) ES-Ray-Convex sets is of much practical interest, as it contains the class of (finite unions of disjunct) convex sets. The above mentioned advantages of ES-estimation to MC-estimation is less prominent for practical cases in which expressions $a(s)$ and $b(s)$ (Definition 3.20) are computationally complex. An intermediate solution in that case could be to approximate, conditionally on $s$, the intersection points $a(s)$ and $b(s)$ using numerical methods.

## 3.8 Robustness bounding methods

The Robustness $R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$ can be intractable to compute directly, as introduced in Section 3.1. The idea of set bounding is to replace $\mathbb{H}(x)$ by a *bounding set*, for which it is easier to compute the Robustness. Such bounding sets lead to upper bounds or lower bounds for $R(x)$ in the following way. The upper bounding sets $\overline{\mathbb{H}}(x)$ are sets that enclose the Happy set $\overline{\mathbb{H}}(x) \supseteq \mathbb{H}(x)$ and the lower bounding sets $\underline{\mathbb{H}}(x)$ are sets that are enclosed by the Happy set $\underline{\mathbb{H}}(x) \subseteq \mathbb{H}(x)$. Three types of bounding sets are studied, that possess analytic properties for efficient probability calculus. These bounding sets are defined by specific $p$-norms, with $p = 1, 2, \infty$:

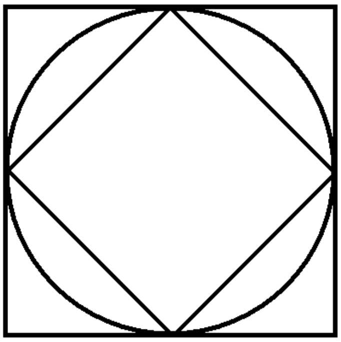


Figure 3.21: Illustration of the Diamond-set, Ball-set and Cube-set in $\mathbb{R}^2$

$$\begin{array}{ll} \text{Diamond-set (1-norm):} & \mathbb{B}^{[1]}(r) = \left\{v \in \mathbb{R}^{\mathrm{N}} \,|\|v\|_1 \leq r\right\} \\ \text{Ball-set (2-norm):} & \mathbb{B}^{[2]}(r) = \left\{v \in \mathbb{R}^{\mathrm{N}} |\, \|v\|_2 \leq r\right\} \\ \text{Cube-set ($\infty$-norm):} & \mathbb{B}^{[\infty]}(r) = \left\{v \in \mathbb{R}^{\mathrm{N}} |\, \|v\|_\infty \leq r\right\} \end{array} \tag{3.95}$$

where $r \in \mathbb{R}$ is called the radius and are illustrated in Figure 3.21. It holds that

$$\mathbb{B}^{[1]}(r) \subset \mathbb{B}^{[2]}(r) \subset \mathbb{B}^{[\infty]}(r) \tag{3.96}$$

since $\|v\|_1 \geq \|v\|_2 \geq \|v\|_\infty$ for $v \in \mathbb{R}^{\mathrm{N}}$ (Randolph, 1968). Moreover, if $a \geq b \geq 0$ then $\Pr_{\boldsymbol{v}}\left(\mathbb{B}^{[p]}(a)\right) \geq \Pr_{\boldsymbol{v}}\left(\mathbb{B}^{[p]}(b)\right)$ for any $p = 1, 2, \infty$ and for any type of probability distribution of $\boldsymbol{v}$. These bounding sets can be used effectively as follows:

- Effective upper bounds, for a given $x$, are defined by the smallest bounding sets $\overline{\mathbb{H}}^{[p]}(x) \supseteq \mathbb{H}(x)$ with $p = 1, 2, \infty$, that enclose the Happy set as tight as possible. The effective upper bounding sets are

  $$\overline{\mathbb{H}}^{[p]}(x) = \mathbb{B}^{[p]}(\overline{r}^{[p]}(x)) \tag{3.97}$$

  where

  $$\overline{r}^{[p]}(x) = \max_{v \in \mathbb{H}(x)} \|v\|_p \tag{3.98}$$

  for $p = 1, 2, \infty$.

- Effective lower bounds, for a given $x$, are defined as follows. The largest bounding set enclosed by the Happy set $\underline{\mathbb{H}}^{[p]}(x) \subseteq \mathbb{H}(x)$ with $p = 1, 2, \infty$, are mainly determined by one of the constraints $\mathrm{L}_s \leq u_s(x, v) \leq \mathrm{H}_s$. Recall that $\mathbb{H}(x)$ is determined by S of such constraint. The largest bounding set inside the Happy set, can be seen either as blowing up the bounding set and notice the first constraint that is hit, or alternatively by determining that constraint $s$ that is in $v$-space closest to the origin. Following the last perspective, under the condition that $E(\boldsymbol{v}) \in \mathbb{H}(x)$ we introduce

  $$\underline{\mathbb{H}}^{[p]}(x) = \mathbb{B}^{[p]}(\underline{r}^{[p]}(x)) \tag{3.99}$$

  where

  $$\underline{r}^{[p]}(x) = \min_{s=1\ldots\mathrm{S}} \left\{ \min_{v \in \mathbb{H}(x)} \|v\|_p \quad \text{s.t. } u_s(x, v) \in \{\mathrm{L}_s, \mathrm{H}_s\} \right\} \tag{3.100}$$

  for $p = 1, 2, \infty$

Notice that these radius optimisation problems can have local optima. It is crucial to find the global optima for radius optimisation problems (3.98) and (3.100) in order to guarantee the correct bounds (as discussed by Olieman and Hendrix, 2005). These Global Optimisation (GO) formulations are only useful if they can be solved *fast enough*, i.e. the bound should be found faster than an accurate estimate. Example 3.8 shows a Happy set with nonlinear boundaries and the corresponding global optimisation problems for finding effective bounds.

**Example 3.8** *Consider the largest enclosed ball in example 2.1. For* $x = [1, 1, 1]^\intercal$*, the two uncertain restrictions are depicted in Figure 3.22 and given by:*

$$\begin{aligned} u_1(x, v) &= -\frac{3}{16}v_1^2 + v_2 + 2 \leq 0 \\ u_2(x, v) &= 1\frac{1}{3}v_1^3 - 4v_1^2 + \frac{2}{3}v_1 + 2 - v_2 \leq 0 \end{aligned}$$

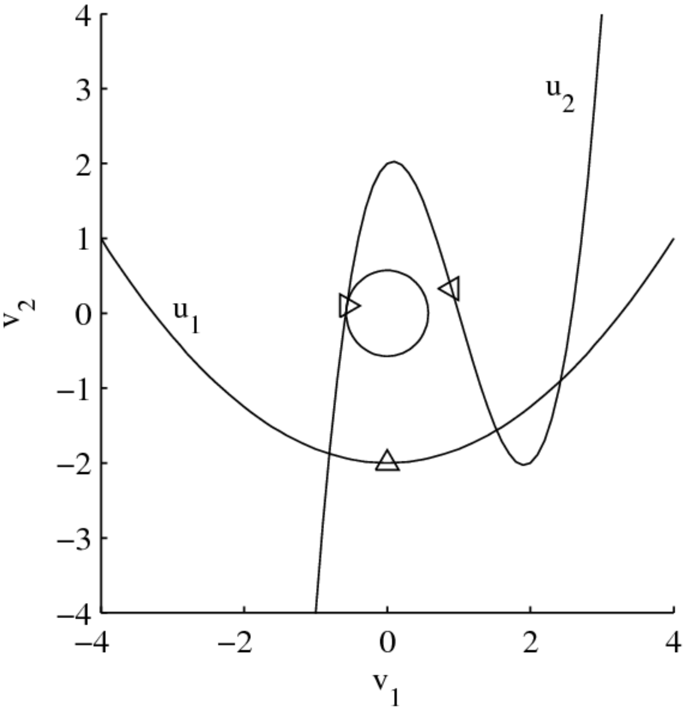


Figure 3.22: Largest inscribed ball in $\mathbb{H}(x)$. The Happy set is the area above u$_1$ and below u$_2$, where u$_1$ and u$_2$ are respectively the graphs defined by $u_1(x, v) = 0$ and $u_2(x, v) = 0$ for given $x = [1, 1, 1]^\intercal$.

*The illustration shows that the inscribed ball first* hits *the restriction* u$_2$ *at the point marked by* $\rhd$*, at the coordinate* $v^* = (-0.565, 0.1)^\intercal$ *where* $u_2(x, v) = 0$*. The coordinate* $v^*$ *corresponds to the* global *solution for* (3.100) *with* $p = 2$ *norm. The two other points, respectively marked with* $\triangle$ *at coordinate* $(0, -2)^\intercal$ *and* $\lhd$ *at coordinate* $(0.9, 0.332)^\intercal$*, correspond to* local *solutions for* (3.100)*.*

For each type of bounding set (i.e. $p = 1, 2, \infty$) and given radius $r \geq 0$, the Robustness bound can be computed analytically, for specific probability distributions of $\boldsymbol{v}$. The idea is that analytic computation is possible if the geometric shape of a bounding set, corresponds to the iso-probability density contour of $\boldsymbol{v}$. The Diamond-set, Ball-set and Cube-set, hold the following properties for computing the corresponding probability mass of the bounding set.

1. Diamond-set (1-norm)

   (a) Let $\boldsymbol{v}_n$, $n = 1, .., \mathrm{N}$, be independently Exponentially distributed with PDF:

$$f(v_n) = \begin{cases} e^{-v_n} & \text{for} \quad v_n \geq 0 \\ 0 & \text{for} \quad v_n < 0 \end{cases} \tag{3.101}$$

The probability mass of the Diamond-set can be computed via Simplex Coordinates Transform (Definition 3.18 on page 59) and it can be shown that

$$\begin{aligned}
\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r)\right\} &= \int\limits_{\mathbb{B}^{[1]}(r)} \prod_{n=1}^{\mathrm{N}} f(v_n) dv \\
&= \int\limits_{\mathbb{B}^{[1]}(r)} e^{-\sum\limits_{n=1}^{\mathrm{N}} v_n} dv \\
&= \int\limits_{\mathbb{S}^{\mathrm{N}-1}} \int\limits_{0}^{r} e^{-r} r^{\mathrm{N}-1} dr ds \\
&= \frac{1}{(\mathrm{N}-1)!} \int\limits_{0}^{r} e^{-r} r^{\mathrm{N}-1} dr = 1 - e^{-r} \sum_{i=0}^{\mathrm{N}-1} \frac{r^i}{i!}
\end{aligned} \tag{3.102}$$

where each step and $\mathbb{S}^{\mathrm{N}-1} = \left\{ s \in \mathbb{R}_{+}^{\mathrm{N}-1} \middle| \sum\limits_{i=1}^{\mathrm{N}-1} s_i \leq 1 \right\}$ are explained in Appendix A.5.

(b) Let $\boldsymbol{v}_n$, $n = 1, .., \mathrm{N}$ be independently two-sided Exponentially distributed with PDF:

$$f(v_n) = \begin{cases} \frac{1}{2} e^{-v_n} & \text{for} \quad v_n \geq 0 \\ \frac{1}{2} e_n^v & \text{for} \quad v_n < 0 \end{cases} \tag{3.103}$$

The probability mass of a Diamond-set is:

$$\begin{aligned}
\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r)\right\} &= \int\limits_{\mathbb{B}^{[1]}(r)} \prod_{n=1}^{\mathrm{N}} f(v_n) dv \\
&= \int\limits_{\mathbb{B}^{[1]}(r)} e^{-\sum\limits_{n=1}^{\mathrm{N}} |v_n|} dv \\
&= 1 - e^{-r} \sum_{i=0}^{\mathrm{N}-1} \frac{r^i}{i!}
\end{aligned} \tag{3.104}$$

where the last step is explained in Appendix A.5.

(c) Let $\boldsymbol{v}_n$, $n = 1, .., \mathrm{N}$ be randomly distributed with $E(\boldsymbol{v}_n) = 0$, $VAR(\boldsymbol{v}_n) = 1$ and $COV(\boldsymbol{v}_n, \boldsymbol{v}_m) = 0$ for all $n \neq m$, then the Markov inequality holds:

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r)\right\} \geq \left(1 - \frac{\mathrm{N}}{r}\right) \tag{3.105}$$

2. Ball-set (2-norm)

(a) Let the elements of random vector $\boldsymbol{v}$ be i.i.d. standard normally distributed. The probability mass of the Ball-set is:

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[2]}(r)\right\} = \Pr\left\{\boldsymbol{\chi}^2(\mathrm{N}) \leq r^2\right\} \tag{3.106}$$

where $\boldsymbol{\chi}^2(\mathrm{N})$ is a $\boldsymbol{\chi}$-square stochastic variate with N degrees of freedom.

(b) Let $\boldsymbol{w} \sim \boldsymbol{\chi}^2(\delta)$, i.i.d. $\boldsymbol{z}_n \sim \mathrm{N}(0,1)$ and $\boldsymbol{v}_n \sim \frac{\boldsymbol{z}_n}{\sqrt{\frac{1}{\delta}\boldsymbol{w}}}$ for $n = 1,..,\mathrm{N}$. I.e. $\boldsymbol{v}_n$ follows a $t$-distribution with $\delta$ degrees of freedom. Somerville (1998) shows that the probability mass of the Ball is:

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[2]}(r)\right\} = \Pr\left\{\boldsymbol{F}(\mathrm{N},\delta) \leq \frac{1}{\mathrm{N}}r\right\} \tag{3.107}$$

(c) Let $\boldsymbol{v}_n$ be randomly distributed with $E(\boldsymbol{v}_n) = 0$, $VAR(\boldsymbol{v}_n) = 1$ and $COV(\boldsymbol{v}_n, \boldsymbol{v}_m) = 0$ for $n \neq m$, then Markov's Inequality (see appendix A.7) applies:

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[2]}(r)\right\} \geq 1 - \frac{\mathrm{N}}{r^2} \tag{3.108}$$

3. Cube-set ($\infty$-norm)

(a) Let $\boldsymbol{v}_n$ be independently distributed random variables with a Cumulative Distribution Function (CDF) $F_n(r) = \Pr\{\boldsymbol{v}_n \leq r\}$ for $n = 1,..,\mathrm{N}$. The probability mass of the cube is:

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[\infty]}(r)\right\} = \prod_{n=1}^{\mathrm{N}} \left(F_n(r) - F_n(-r)\right)) \tag{3.109}$$

(b) Let $\boldsymbol{v}_n$ be randomly distributed with $E(\boldsymbol{v}_n) = 0$, $VAR(\boldsymbol{v}_n) = 1$ and $COV(\boldsymbol{v}_n, \boldsymbol{v}_m) = 0$ for all $n \neq m$, then the Chebyshev Inequality holds:

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[\infty]}(r)\right\} \geq \left(1 - \frac{1}{r^2}\right)^{\mathrm{N}} \tag{3.110}$$

The probabilities of the Diamond-set, Ball-set and Cube-set are all increasing functions of the radius. Therefore, Robustness bound maximisation can be based on finding the largest radius of the bounding set. However, Definition (3.99) and (3.100) only apply for $E(\boldsymbol{v}) \in \mathbb{H}(x)$. The idea of maximising the lower bound indirectly via the radius, is generalised by redefining the radius for all $x \in \mathbb{X}$:

$$\underline{r}^{[p]}(x) = \begin{cases} \displaystyle\min_{s=1\ldots\mathrm{S}} \left\{ \min_{v \in \mathbb{H}(x)} \|v\|_p \quad \text{s.t. } u_s(x,v) \in \{\mathrm{L}_s, \mathrm{H}_s\} \right\} & \text{if } E(\boldsymbol{v}) \in \mathbb{H}(x) \\ -1\cdot \displaystyle\min_{s=1\ldots\mathrm{S}} \left\{ \min_{v \in \mathbb{H}(x)} \|v\|_p \quad \text{s.t. } u_s(x,v) \in \{\mathrm{L}_s, \mathrm{H}_s\} \right\} & \text{if } E(\boldsymbol{v}) \notin \mathbb{H}(x) \end{cases} \tag{3.111}$$

If $E(\boldsymbol{v}) \notin \mathbb{H}(x)$, then the (negative) radius can be interpreted as the shortest p-norm distance between $E(\boldsymbol{v})$ and the point closest to $E(\boldsymbol{v})$ in the Happy set. Hence, maximising the radius $\underline{r}^{[p]}(x)$ is equivalent to searching for a design $x$ such that $E(\boldsymbol{v}) \in \mathbb{H}(x)$. From a practical point of view, finding $x$ such that $E(\boldsymbol{v}) \in \mathbb{H}(x)$, is relevant because of the following. From the results in Appendix A.6 follows that, for many types of distributions and convex Happy sets, holds that $\Pr(\mathbb{H}(x)) \geq \Pr(\mathbb{H}(y))$ if $E(\boldsymbol{v}) \in \mathbb{H}(x)$, $E(\boldsymbol{v}) \notin \mathbb{H}(y)$ and $\Pr(\mathbb{H}(x)) \geq \frac{1}{2}$. In Figures 3.25, 3.28 and 3.29, $E(\boldsymbol{v}) \notin \mathbb{H}(x^{[0]})$ and $E(\boldsymbol{v}) \in \mathbb{H}(x^{*})$ are illustrated, where the Happy set is a polytope.

Example 3.9 gives an illustration of Robustness Lower bound optimisation.

**Example 3.9** *Consider Robustness function $R(x)$ defined by (3.5) of Example 3.1, where $E(\boldsymbol{v}_n) = 0$ and $VAR(\boldsymbol{v}_n) = 1$ for $n = 1, 2$ in the Gaussian situation. The left-hand side of Figure 3.23 shows the mesh surface of the (level 2) largest radius of the Ball in the Happy set. The right-hand side of Figure 3.23 shows the corresponding probability mass in the Gaussian situation. The left-hand side of Figure 3.24 shows the mesh surface of the (level 2) largest radius of the Cube in the Happy set. The right-hand side of Figure 3.24 shows the corresponding probability mass in the Gaussian situation. We study the optimisation of these bounds, similar to the optimisation of the MC estimate function of Example 3.1: The starting point is $x^{[0]} = (0.5, -0.3)$; FD step-sizes are $\Delta \in \left\{\left(\frac{1}{2}\right)^{10}, \left(\frac{1}{2}\right)^{9}, ..., \left(\frac{1}{2}\right)^{1}, \left(\frac{1}{2}\right)^{0}\right\}$; All other parameters are kept at the Matlab default values. Table 3.6 gives the optimal values of $\underline{r}^{[p]*} = \max_{-1 \leq x_1, x_2 \leq 1} \underline{r}^{[p]}(x)$, $p = 2, \infty$ to which the SQP algorithm converges, where vector $x^{[p]*}$ is the corresponding design. The best result for the largest Ball set in the Happy set was found using a step size of $\Delta = \left(\frac{1}{2}\right)^{10}$. The corresponding optimisation results are shown in Figure 3.28. The best result for the largest Cube in the Happy set was found with $\Delta = \left(\frac{1}{2}\right)^{10}$. The corresponding optimisation results are shown in Figure 3.29.*

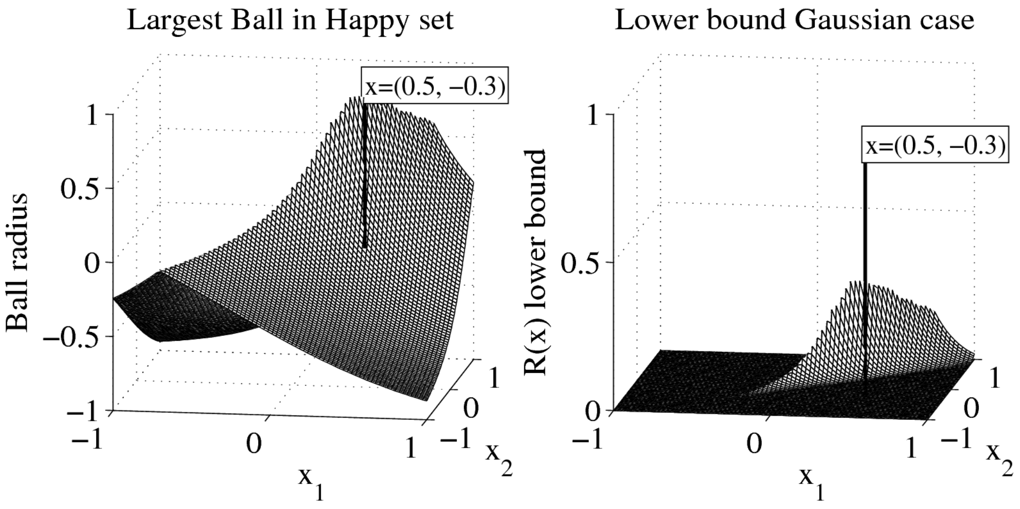


Figure 3.23: Illustration of Robustness bounds, based on largest Ball in Happy set

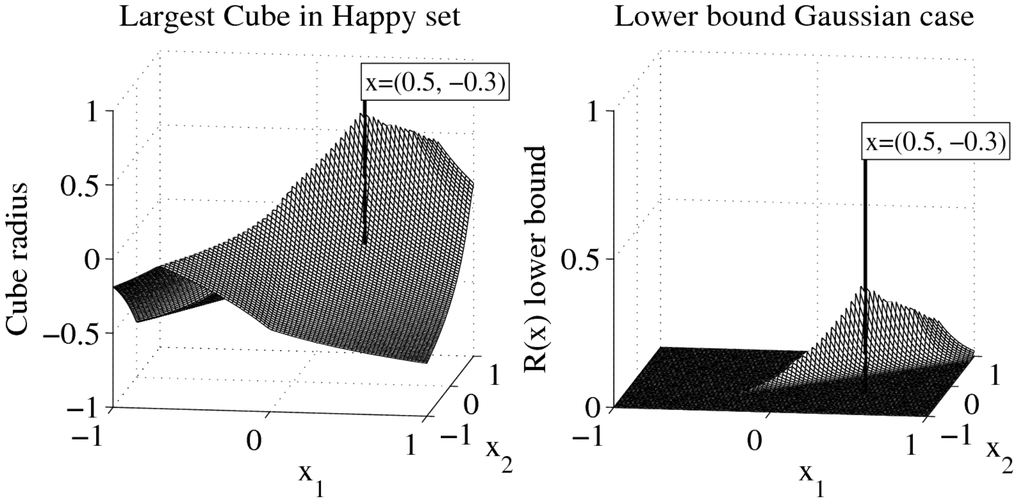


Figure 3.24: Illustration of Robustness bounds, based on largest Cube in Happy set

Example 3.10 gives an illustration of Robustness upper bound optimisation.

**Example 3.10** *Consider the Robustness function $R(x)$ of Example 3.9. From* (3.98) *follows that such problem can be unbounded and result in $\overline{r}^{[p]}(x^*) = \infty$. To illustrate such situation graphically, the mesh surface of the inverse $\frac{1}{\overline{r}^{[p]}(x)}$ is considered instead.*

*Optimisation of the Ball and Cube upper bounds, is studied similar to the optimisation study of the MC estimate function of Example 3.1: The starting point is $x^{[0]} = (0.5, -0.3)$; FD step-sizes are $\Delta \in \left\{\left(\frac{1}{2}\right)^{10}, \left(\frac{1}{2}\right)^{9}, ..., \left(\frac{1}{2}\right)^{1}, \left(\frac{1}{2}\right)^{0}\right\}$; All other parameters are kept at Matlab default values. Table 3.7 gives the optimal values of $\overline{r}^{[p]*} = \max_{-1 \le x_1, x_2 \le 1} \overline{r}^{[p]}(x)$, $p = 2, \infty$ to which the SQP algorithm converges. Vector $x^{[p]*}$ is the corresponding optimal design.*

*The best result for the smallest Ball and Cube around the Happy set was found with $\Delta = \left(\frac{1}{2}\right)^{10}$. The corresponding optimisation results are shown in Figure 3.28 and 3.29.*

It can be concluded that optimisation of Robustness upper bounds is not satisfactory for the following reasons:

- The upper bounds are not tight enough: The Robustness upper bound at the starting point $x^{[0]}$ has already the maximum value 1.
- Maximisation of the upper bounds is equivalent to finding an $x^*$ for which the Happy set is unbounded. However, an unbounded Happy set is not necessarily related to a high Robustness.

Table 3.6: SQP Optimisation results largest ball ($\underline{r}^{[2]*}$) and largest cube ($\underline{r}^{[\infty]*}$) inscribed by Happy set, for given FD step size$\Delta$

| | Ball | | | Cube | | |
|---|---|---|---|---|---|---|
| $\Delta$ | $\underline{r}^{[2]*}$ | $\Pr\left(\mathbb{B}^{[2]}(\underline{r}^{[2]*})\right)$ | $\tilde{R}\left(x^{[2]*}\right)$ | $\underline{r}^{[\infty]*}$ | $\Pr\left(\mathbb{B}^{[\infty]}(\underline{r}^{[\infty]*})\right)$ | $\tilde{R}\left(x^{[\infty]*}\right)$ |
| starting points at $x^{[0]} = (0.5, -0.3)$: | | | | | | |
| | -0.497 | NA | 0.294 | -0.3750 | NA | 0.294 |
| $\left(\frac{1}{2}\right)^{10}$ | 0.873 | 0.317 | 0.715 | 0.737 | 0.290 | 0.695 |
| $\left(\frac{1}{2}\right)^{9}$ | 0.873 | 0.317 | 0.715 | 0.727 | 0.284 | 0.690 |
| $\left(\frac{1}{2}\right)^{8}$ | 0.867 | 0.313 | 0.688 | 0.734 | 0.288 | 0.698 |
| $\left(\frac{1}{2}\right)^{7}$ | 0.664 | 0.198 | 0.679 | 0.729 | 0.285 | 0.699 |
| $\left(\frac{1}{2}\right)^{6}$ | 0.681 | 0.207 | 0.689 | 0.736 | 0.290 | 0.698 |
| $\left(\frac{1}{2}\right)^{5}$ | 0.678 | 0.206 | 0.687 | 0.354 | 0.077 | 0.437 |
| $\left(\frac{1}{2}\right)^{4}$ | 0.551 | 0.141 | 0.590 | 0.379 | 0.087 | 0.472 |
| $\left(\frac{1}{2}\right)^{3}$ | 0.389 | 0.073 | 0.513 | 0.377 | 0.086 | 0.471 |
| $\left(\frac{1}{2}\right)^{2}$ | 0.209 | 0.022 | 0.416 | 0.305 | 0.058 | 0.404 |
| $\left(\frac{1}{2}\right)^{1}$ | 0.317 | 0.049 | 0.379 | 0.373 | 0.085 | 0.470 |
| $\left(\frac{1}{2}\right)^{0}$ | 0.604 | 0.167 | 0.425 | 0.443 | 0.117 | 0.425 |
| Average computation time and average estimated standard error per function evaluation are: | | | | | | |
| time | 0.0015 | 0.002 | 151.001 | 0.0014 | 0.0018 | 153.879 |
| $\hat{se}$ | NA | 0 | 0.0005 | NA | 0 | 0.0005 |

Comments on Table 3.6:

1. $\underline{r}^{[p]*}(x) = \max\limits_{-1 \leq x_1, x_2 \leq 1} \underline{r}^{[p]}(x)$, with $p = 2, \infty$
2. $x^{[p]*} = \arg\max\limits_{-1 \leq x_1, x_2 \leq 1} \underline{r}^{[p]}(x)$, with $p = 2, \infty$
3. $\tilde{R}(x^*)$ is an MC estimate of $R(x^*)$, with $\hat{se}\left(\tilde{\boldsymbol{R}}(x^*)\right) < 0.0005$ (See Appendix A.3).

Table 3.7: SQP Optimisation results smallest ball ($\overline{r}^{[2]*}$) and smallest cube ($\overline{r}^{[\infty]*}$) enclosing Happy set, for given $\Delta$

| | Ball | | | Cube | | |
|---|---|---|---|---|---|---|
| $\Delta$ | $\overline{r}^{[2]*}$ | $\Pr\left(\mathbb{B}^{[2]}(\overline{r}^{[2]*})\right)$ | $\tilde{R}\left(x^{[2]*}\right)$ | $\overline{r}^{[\infty]*}$ | $\Pr\left(\mathbb{B}^{[\infty]}(\overline{r}^{[\infty]*})\right)$ | $\tilde{R}\left(x^{[\infty]*}\right)$ |
| starting points at $x^{[0]} = (0.5, -0.3)$: | | | | | | |
| | 9.430 | 1 | 0.294 | 8.778 | 1 | 0.294 |
| $\left(\frac{1}{2}\right)^{10}$ | $\infty$ | 1 | 0.3030 | $\infty$ | 1 | 0.2930 |
| $\left(\frac{1}{2}\right)^{9}$ | $\infty$ | 1 | 0.3080 | $\infty$ | 1 | 0.2980 |
| $\left(\frac{1}{2}\right)^{8}$ | $\infty$ | 1 | 0.3450 | $\infty$ | 1 | 0.2930 |
| $\left(\frac{1}{2}\right)^{7}$ | $\infty$ | 1 | 0.3030 | $\infty$ | 1 | 0.3040 |
| Average computation time and average estimated standard error per function evaluation are: | | | | | | |
| time | 0.0015 | 0.002 | 151.001 | 0.0014 | 0.0018 | 153.879 |
| $\hat{se}$ | NA | 0 | 0.0005 | NA | 0 | 0.0005 |

Comments on Table 3.7:

1. $\overline{r}^{[p]*}(x) = \max\limits_{-1 \leq x_1, x_2 \leq 1} \overline{r}^{[p]}(x)$, with $p = 2, \infty$
2. $x^{[p]*} = \arg\max\limits_{-1 \leq x_1, x_2 \leq 1} \overline{r}^{[p]}(x)$, with $p = 2, \infty$
3. $\tilde{R}(x^*)$ is an MC estimate of $R(x^*)$, with $\hat{se}\left(\tilde{\boldsymbol{R}}(x^*)\right) < 0.0005$ (See Appendix A.3).

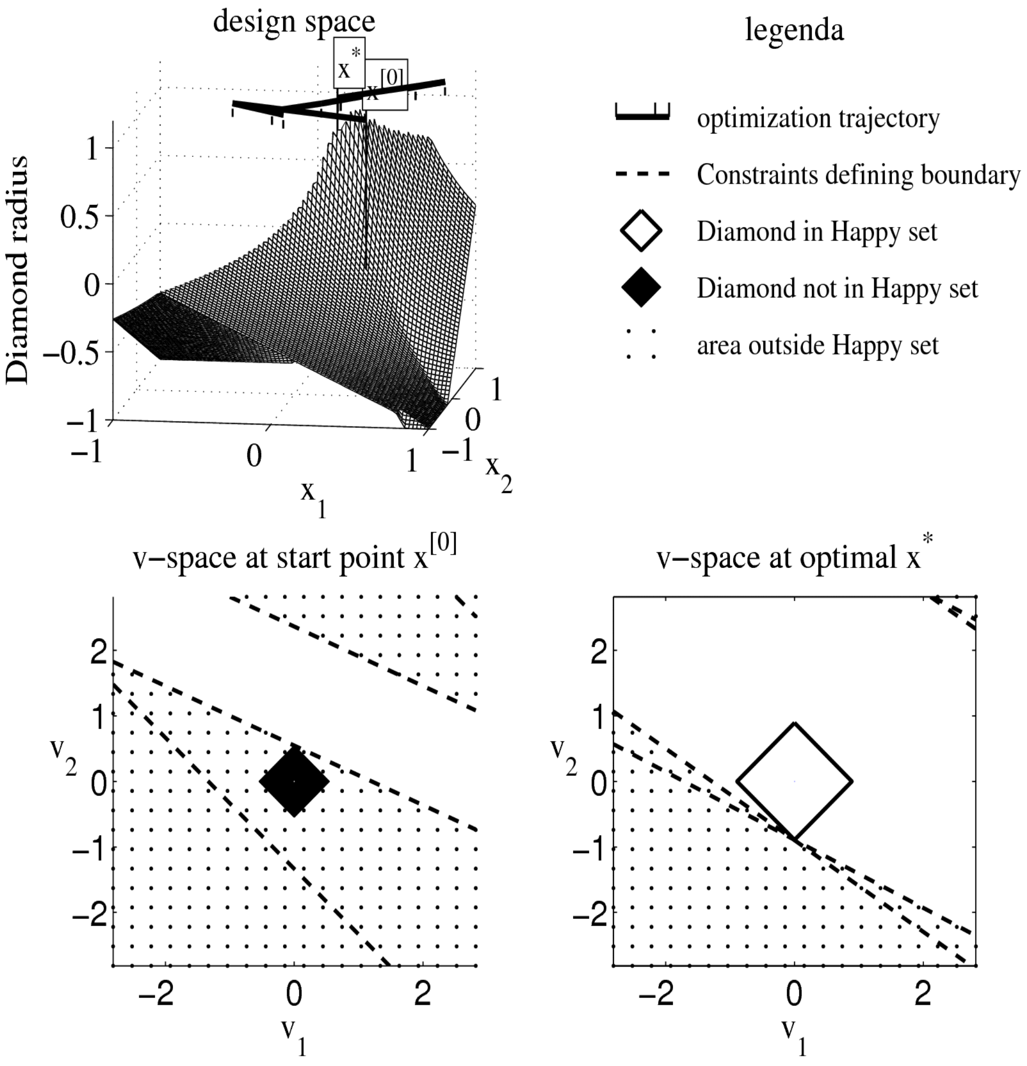


Figure 3.25: SQP Largest Diamond in Happy set optimisation results; $\Delta = \left(\frac{1}{2}\right)^{10}$

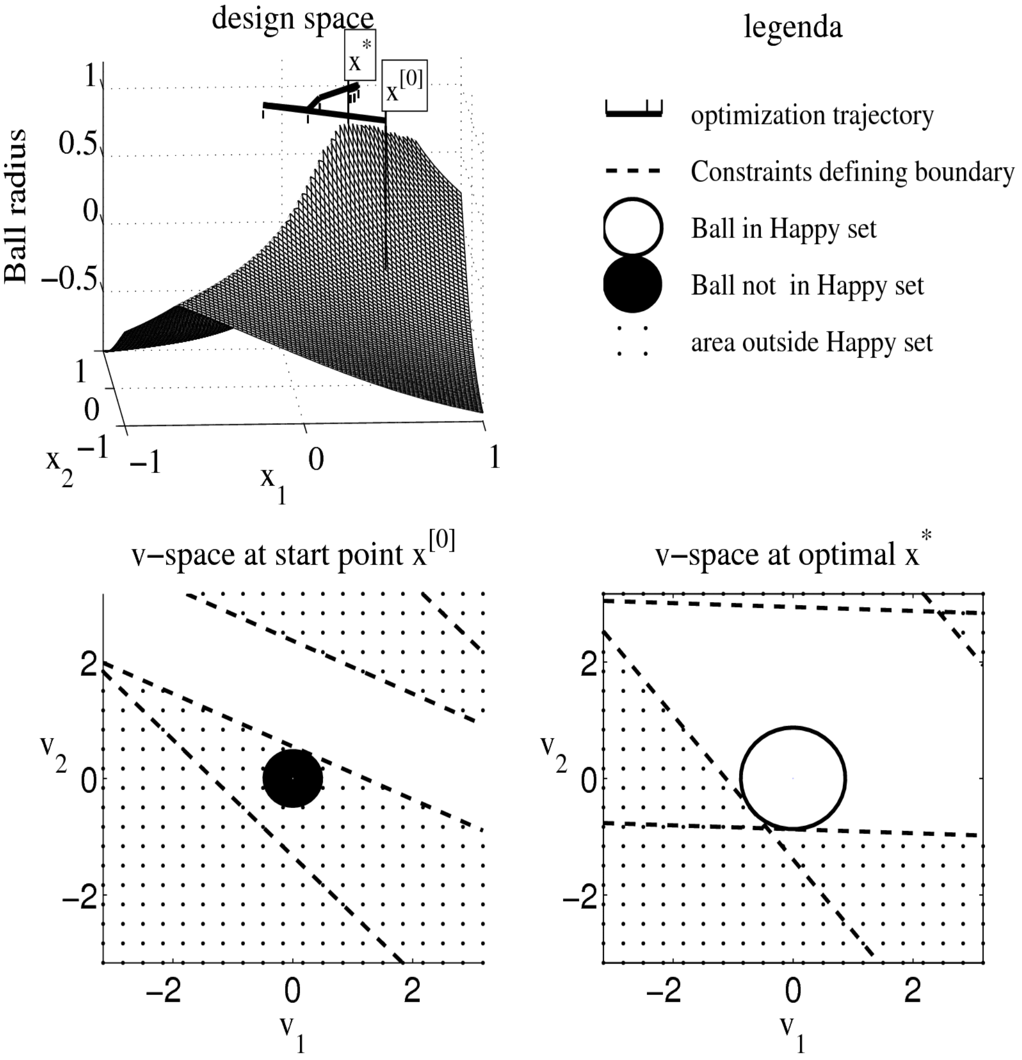


Figure 3.26: SQP Largest Ball in Happy set optimisation results; $\Delta = \left(\frac{1}{2}\right)^{10}$

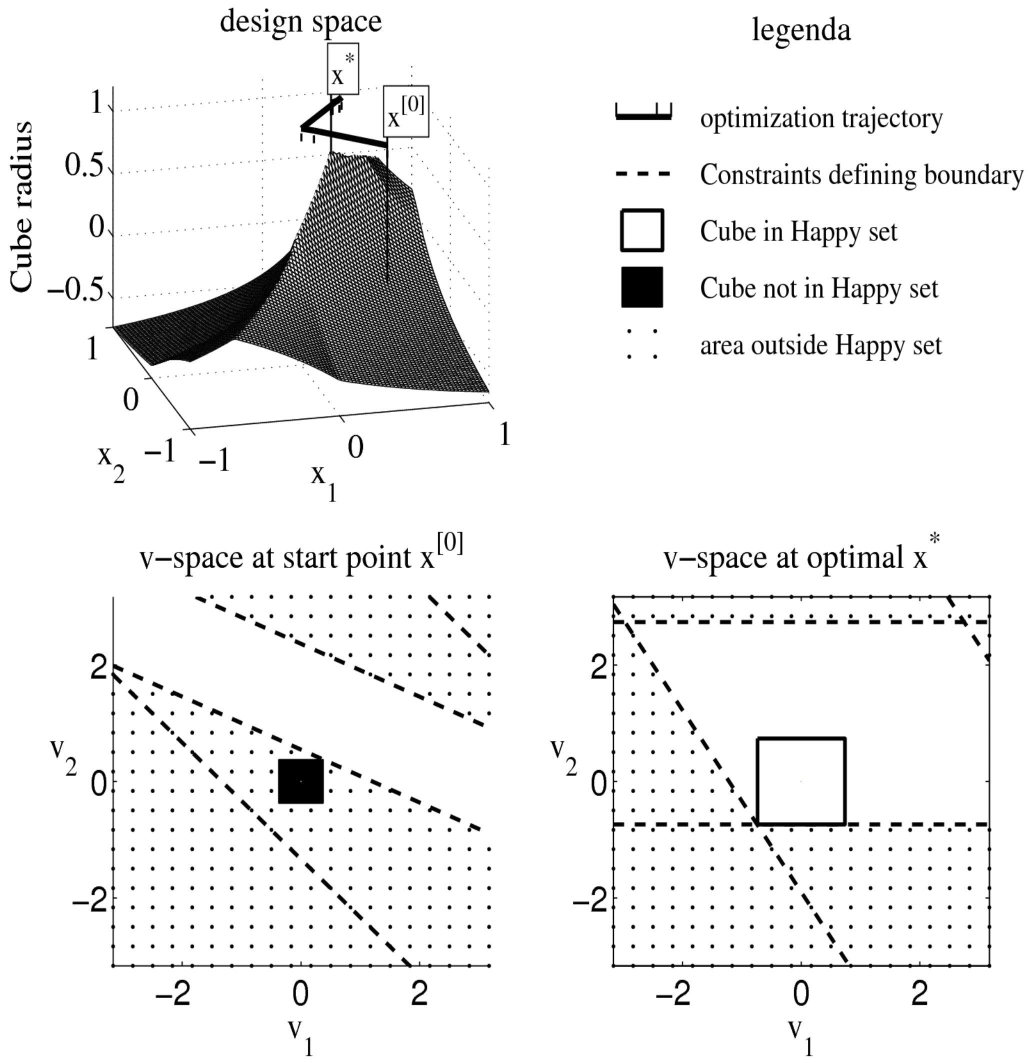


Figure 3.27: SQP Largest Cube in Happy set optimisation results; $\Delta = \left(\frac{1}{2}\right)^{10}$

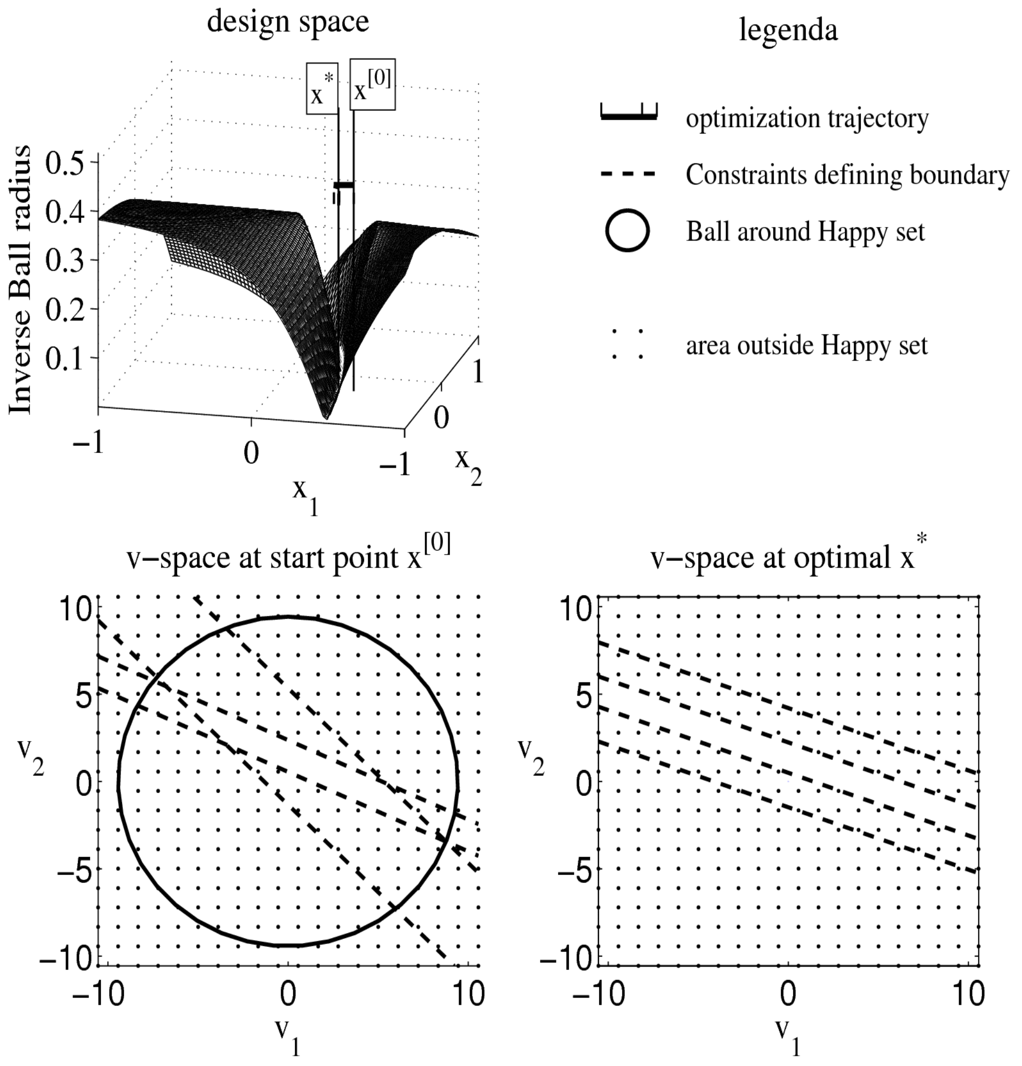


Figure 3.28: SQP maximisation of smallest Ball around Happy set ; $\Delta = \left(\frac{1}{2}\right)^{10}$

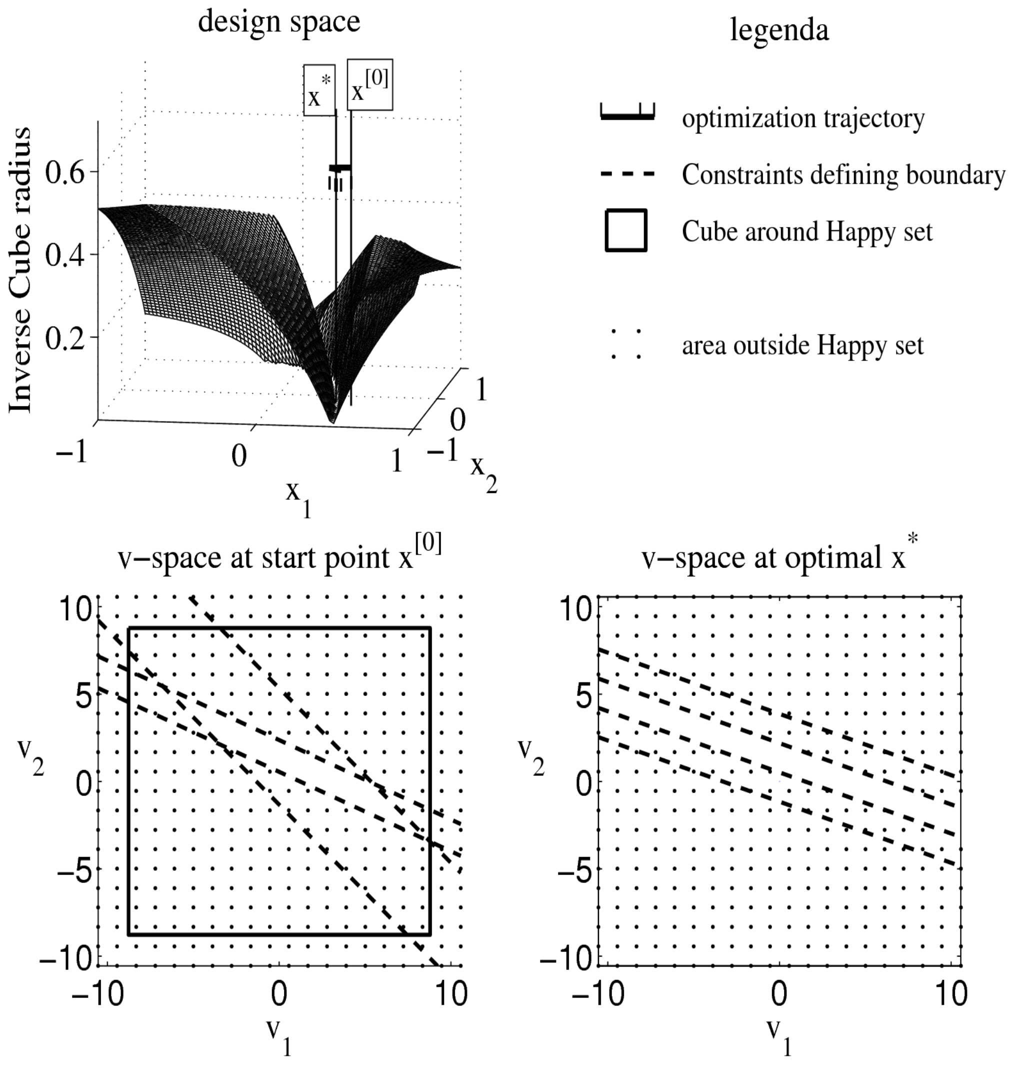


Figure 3.29: SQP maximisation of smallest Cube around Happy set ; $\Delta = \left(\frac{1}{2}\right)^{10}$

## 3.9 Concluding remarks

Robustness estimation methods have been discussed in this chapter. Two types of Robustness estimation methods are distinguished: sampling methods and bounding methods. The Robustness estimation methods that are based on sampling are the MC, SMC, N-1MC, DS and ES estimation method. The Robustness bounding methods are based on sets inscribed by the Happy set or on sets enclosing the Happy set and respectively lead to Robustness lower bounds and Robustness upper bounds. The inscribed sets and enclosing sets have specific geometric shapes, such that their probability masses are relatively easy to compute in comparison to computing the probability mass of the Happy set directly. Three sets with a specific geometric shape are used and are respectively called the Diamond set (based on the 1-norm), the Ball set (based on the 2-norm) and the Cube set (based on the $\infty$-norm).

The Robustness estimation methods are applied in case studies as discussed in Chapter 5. The research related to solving relatively large-scale Robustness optimisation problems of the case studies, resulted in computation methods that improve the efficiency and effectiveness of the RP methods and are discussed in Chapter 4. Furthermore, the case studies are used to compare the RP methods and the methodology for comparing the performance of the RP methods is also discussed in Chapter 4.

# Chapter 4
# Robustness Computation and Comparison

## 4.1 Outline

Robustness estimation methods have been defined and described in Chapter 3. The focus of Chapter 4 is about how to compute Robustness estimates effectively and efficiently. Robustness estimation effectiveness is the level of accuracy of the estimate, i.e. the absolute value of the difference between the (unknown) Robustness and the estimated Robustness. Robustness computation efficiency is the computation time required to estimate the Robustness at a predefined effectiveness level.

In elaboration of cases, which are discussed in Chapter 5, it became clear that increasing $v$-space dimension makes the Robustness estimation methods less effective and efficient. In this chapter, so-called *dimension reduction* methods are introduced for transforming the Happy set into a lower dimensional space, leading to a better Robustness computation performance. Section 4.2 gives an intuition and mathematical elaboration on the effect of the dimension N of the Happy set on the performance of Robustness estimation. In Section 4.3 two methods for dimension reduction, respectively called Happy set *decomposition* and Happy set *compression*, are introduced. General expressions for dimension reduction of polyhedral Happy sets are given in Section 4.4.

Solving the Robustness optimisation problems of case studies, made it clear that standard optimisation software has difficulties to start optimisation iterations, if the estimates of $R(x)$ are zero for $x$ in a neighbourhood around the initial starting point. In Section 4.5 the so-called Warm Start method is discussed as a practical approach for this optimisation start-up problem.

The case studies of Chapter 5 are used to compare the RP methods. The methodology for comparing the performance of the RP methods is given in Section 4.6.

## 4.2 Intuition on Dimension Reduction

The dimension (N) of the Happy set influences the performance of the Robustness sampling and bounding methods. This section illustrates that dimension reduction of the Happy set leads to *equal* or *tighter* Robustness bounds, i.e. dimension reduction leads to more effective bounds. Moreover, dimension reduction of the Happy set makes the Robustness sampling estimation method more accurate (lower standard error), which reduces the required number of samples to reach a predefined level of accuracy. A smaller number of samples, means less computations and is therefore associated to increasing efficiency.

Two dimension reduction approaches for improving the performance are considered: Happy set *decomposition* and Happy set *compression*. These two concepts are introduced intuitively by three examples:

1. Happy set decomposition in a bounding context is illustrated by Example 4.1.

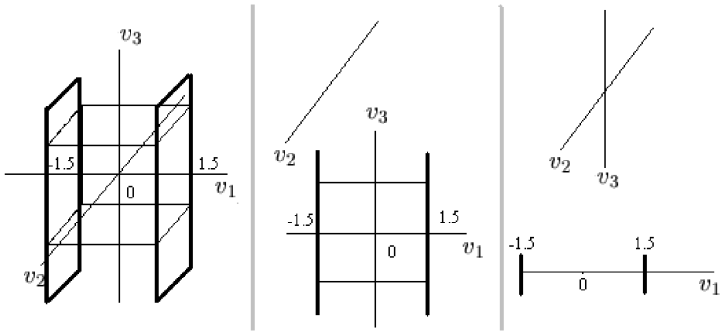


Figure 4.1: Decomposition and Bounding example for N = 3, N = 2 and N = 1

2. Happy set compression in a bounding context is illustrated by Example 4.2.
3. Dimension reduction -either via decomposition or compression- in the sampling context is illustrated by Example 4.3.

After these three examples, a more precise description of the concepts of decomposition and compression is given in Section 4.3.

**Example 4.1 (Decomposition)** *Chapter 3 has shown that the probability mass of an inscribed hyper cube in the Happy set is a lower bound of the Robustness. Consider Figure 4.1. A hyper cube with radius* 1.5 *around the origin is used for finding a lower bound for the Robustness of the Happy set defined by two half-spaces* $\mathbb{H}(x) = \left\{ v \in \mathbb{R}^N | -1.5 \leq \sum_{n=1}^{\mathrm{N}} v_n x_n \leq 1.5 \right\}$ *where* $x_1 = 1$*,* $x_i = 0$ *for* $i = 2, .., I = \mathrm{N}$ *and* $\boldsymbol{v}_n \sim N(0,1)$ *for* $n = 1, .., \mathrm{N}$*. Notice that the probability mass (i.e. the Robustness) of the Happy set is* $\Phi(1.5) - \Phi(-1.5) \approx 0.87$ *independently of* N*. The probability mass of the cube goes down exponentially with the dimension:* $(\Phi(1.5) - \Phi(-1.5))^N$*, i.e. the effectiveness of the bound goes down exponentially as a function of the dimension* N*. In this particular example one can make use of the independency of* $\boldsymbol{v}_n$*. We will call this* Happy set decomposition by independency*. We split the Happy set into two lower dimensional parts, one Happy set corresponding to* $\boldsymbol{v}_1$ *and one Happy set corresponding to all other* $\boldsymbol{v}_n$ *for* $n = 2, .., \mathrm{N}$*. This simple example has an analytical solution: The Robustness of the first Happy set can be computed by the univariate standard normal CDF function and has value* $\Phi(1.5) - \Phi(-1.5) \approx 0.87$*. Since the second Happy set is unconstrained, it has Robustness 1. The overall Robustness estimation by decomposition is approximately* 0.87*. This method is discussed in more detail in Section 4.3.*

**Example 4.2 (Compression)** *Let* $\mathbb{H}(x) = \left\{ v \in \mathbb{R}^3 | \sum_{n=1}^{3} v_n x_n \geq -4.5 \right\}$ *be the Happy set of interest, with* $x_1 = x_2 = x_3 = 1$ *and* $\boldsymbol{v}_n \sim N(0,1)$ *for* $n = 1, .., 3$*. The left-hand side of Figure 4.2 illustrates the corresponding half space, where the Happy set is an unbounded subspace of* $\mathbb{R}^{\mathrm{N}}$*, with* N*=3. Since the random variables are normally distributed, their sum is also normally distributed. This feature can be exploited to* compress *the Happy set into a lower dimensional Happy set, by redefining the stochastic model. The random*

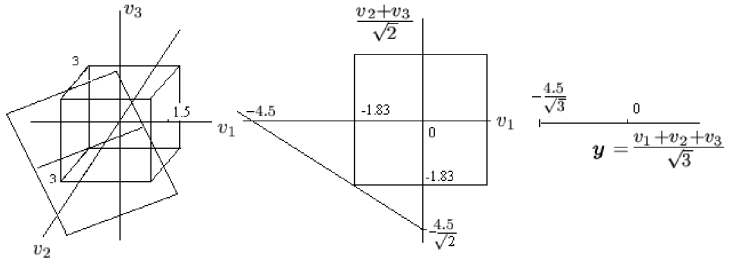


Figure 4.2: Compression and Bounding example for N = 3, N = 2 and N = 1

*variables $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ can be combined to a new random variable $\frac{\boldsymbol{v}_2+\boldsymbol{v}_3}{\sqrt{2}} \sim N(0,1)$. This way, a new Happy set can be expressed as a subspace of $\mathbb{R}^{\mathrm{N}}$, with $\mathrm{N}$=2 as is illustrated in the middle part of Figure 4.2. The compression of $\mathrm{N}$=3 to $\mathrm{N}$=2, positively affects the bounding efficiency in two ways:*

1. *Dimension reduction by itself results in an effectiveness increase: A $1.5 \times 1.5$ cube in $\mathrm{N} = 2$ has a larger probability mass than a cube of $1.5 \times 1.5 \times 1.5$ in $\mathrm{N} = 3$ as illustrated in the previous example.*

2. *Additionally, the radius of the the largest cube inside the Happy set increases from 1.5 to 1.83, thus making the bound even more effective.*

*In this example, there exists an exact solution via compression with $\frac{1}{\sqrt{3}}(\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3) \sim N(0,1)$ and gives that the Robustness can be expressed as: $R(x) = \Phi(\frac{4.5}{\sqrt{3}}) \approx 0.995$.*

**Example 4.3 (Dimension reduction efficiency)** *Chapter 3 has shown that the accuracy of the DS Robustness estimator tends to the efficiency of the MC estimator if the DS estimator has a high probability on the two outcomes in $\{0,1\}$. The following example shows a case where the DS method tends to $\{0,1\}$ outcomes, with increasing dimension. Figure 4.3 shows the Directional Sampling method estimating the Robustness of Happy set*

$$\mathbb{H}(x) = \left\{ v \in \mathbb{R}^{\mathrm{N}} | -1.5 \leq \sum_{n=1}^{\mathrm{N}} v_n x_n \leq 1.5 \right\} \tag{4.1}$$

*where, $x_1 = 1$, $x_i = 0$, for $i = 2,..,I = \mathrm{N}$ and $\boldsymbol{v}_n \sim N(0,1)$ for $n = 1,..,\mathrm{N}$. For this case, Directional Sampling works as follows. Let $\boldsymbol{s}$ be a uniformly distributed random vector on the unit sphere. Let $r(s)$ be the distance between the origin and the point where the vector through $s$ intersects the bounding plane of the Happy set, i.e. the length of the vector in Figure 4.3. From Deák's estimate function (3.69) on page 52 follows that the unbiased Robustness estimate is:*

$$\overset{ds}{r}(s,x) = \Pr\left\{\boldsymbol{\chi}^2(\mathrm{N}) \leq r^2(s)\right\} \tag{4.2}$$

*where $\boldsymbol{\chi}^2(\mathrm{N})$ is a chi-square distributed random variable with $\mathrm{N}$ degrees of freedom. In Appendix A.8 is shown that if $\mathrm{N} \to \infty$ then $\Pr\left\{0 < \overset{ds}{r}(\boldsymbol{s},x) < 1\right\} \to 0$. This means that the standard error of the DS estimator approaches the standard error of the MC estimator.*

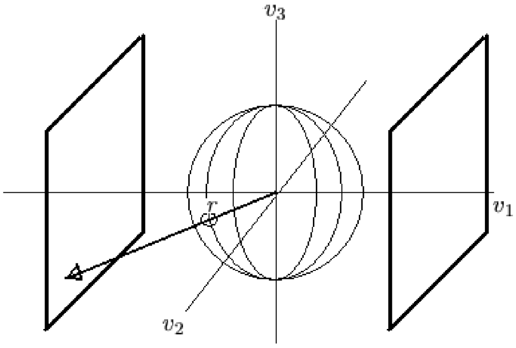


Figure 4.3: Directional Sampling for estimating Robustness

The conjecture based on these examples, is that dimension reduction can improve the performance of Robustness estimation methods. The concept of dimension reduction is mathematically elaborated in the next section.

## 4.3 Happy set Dimension Reduction

The Happy set is defined as $\mathbb{H}(x) = \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \mathrm{L}_s \leq u_s(x, v) \leq \mathrm{H}_s,\ s = 1, .., \mathrm{S} \right\}$ and defines the Robustness $R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$. We will say that such Happy set has dimension N and is based on S uncertain restrictions, where $u_s(x, v)$ is called an *uncertain object property* and $u_s(x, \boldsymbol{v})$ is called a *random object property.* The following is a concise introduction to two dimension reduction concepts called decomposition and compression. Both decomposition and compression are *reformulations* of the Happy set $\mathbb{H}(x)$.

- **Decomposition** is the method to search for a partition of the index set $\mathbb{S} = \{1, .., \mathrm{S}\}$, based on independency of random object properties $u_s(x, \boldsymbol{v})$ that describe the Happy set. The partition leads to an equivalent definition for Robustness, based on a class of lower dimensional Happy sets. Let the mutually disjoint sets $\mathbb{G}_g$ for $g = 1, .., \mathrm{G}$ be the partition-sets of the partition of $\mathbb{S}$, with $\bigcup\limits_{g=1}^{\mathrm{G}} \mathbb{G}_g = \mathbb{S}$. The following rule applies for each partition-set $g$ with $g = 1, .., \mathrm{G}$: Each random object property $u_s(x, \boldsymbol{v})$ with $s \in \mathbb{G}_g$ is independent with respect to each random object property $u_t(x, \boldsymbol{v})$ with $t \in \mathbb{S} \backslash \mathbb{G}_g$. Section 4.3.1 shows that $u_s(x, \boldsymbol{v})$ and $u_t(x, \boldsymbol{v})$ are independent if they do not depend on common elements of $\boldsymbol{v}$. The partition of $\{1, .., \mathrm{S}\}$ into inter-independent sets of random object properties corresponds to decomposing Happy set $\mathbb{H}(x) \subseteq \mathbb{R}^{\mathrm{N}}$ into G Happy sets $\mathbb{H}_g(x) \subseteq \mathbb{R}^{\mathrm{M}_g}$ with $g = 1, .., \mathrm{G}$, each having a dimension $\mathrm{M}_g$ such that $\sum\limits_{g=1}^{\mathrm{G}} \mathrm{M}_g = \mathrm{N}$. The elements of vector $v \in \mathbb{H}(x)$ are partitioned correspondingly, i.e. the lower dimensional Happy set has elements $v^{[g]} \in \mathbb{H}_g(x)$ and each element of vector $v$ corresponds uniquely to one element of one of the vectors $v^{[1]}, .., v^{[g]}, .., $ or $v^{[\mathrm{G}]}$. The Robustness of the Happy set is equivalent to the product of the Robustness of the Happy sets defined by the random object properties of each partition-set separately, since two random object properties in different partition-sets are independent: $R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\} = \prod_{g=1}^{\mathrm{G}} \Pr\left\{\boldsymbol{v}^{(g)} \in \mathbb{H}_g(x)\right\}$. Consequently, if there exists such a partition, then the dimension of each $\mathbb{H}_g(x)$ is strictly smaller than the original dimension N. The intuition given in Section 4.2 makes it plausible that the efficiency of the RP methods for estimating $R(x)$, will improve for increasing G. This idea is worked out in more detail in Section 4.3.1.

- **Compression** reduces the dimension of the Happy set by defining an equivalent Happy set, based on a stochastic vector $\boldsymbol{y}$ lower in dimension than $\boldsymbol{v}$. Compression is an option in the situation that the stochastic model allows the definition of a function of random variables with a lower dimensional domain, which is equivalent to the original function of random variables. For example, the sum of $k$ normally distributed random variables is a univariate normally distributed random variable and the sum of $k$ i.i.d. chi-square random variables is chi-square distributed with $k$ degrees of freedom. In both examples, replacing the original $k$ variables, by the model of the equivalent univariate random variable, compresses the Happy set dimension from N to $\mathrm{N} - k + 1$.

### 4.3.1 Happy set Decomposition

The partition of $\mathbb{S} = \{1, .., \mathrm{S}\}$ is based on independency of uncertain object properties of the Happy set for fixed $x$. Consider two uncertain object property indices $s, t \in \mathbb{S}$, $s \neq t$. For a given $x$, $u_s(x, \boldsymbol{v})$ and $u_t(x, \boldsymbol{v})$ are independent if

$$\begin{aligned} &\Pr_{\boldsymbol{v}}\left(\{v \,|\mathrm{L}_s \leq u_s(x,v) \leq \mathrm{H}_s,\, \mathrm{L}_t \leq u_t(x,v) \leq \mathrm{H}_t\}\right) \\ =&\Pr_{\boldsymbol{v}}\left(\left\{v \,\middle|\, \mathrm{L}_s \leq u_s(x,v) \leq \mathrm{H}_s \right\}\right) \Pr_{\boldsymbol{v}}\left(\left\{v \,\middle|\, \mathrm{L}_t \leq u_t(x,v) \leq \mathrm{H}_t \right\}\right) \end{aligned} \tag{4.3}$$

for all $\mathrm{H}_s, \mathrm{H}_t \in \overline{\mathbb{R}}$. Since all elements of random vector $\boldsymbol{v}$ are by definition independent, a sufficient condition for independency of random object properties $s$ and $t$, is that the underlying functions $u_s(x, v)$ and $u_t(x, v)$ do not have any element $v_n$ in common.

**Example 4.4** *Let the uncertain object properties be:*

$$\begin{aligned} u_1(x,v) &= 2x_1v_1 + x_2v_2 + x_3v_3v_4 \\ u_2(x,v) &= x_2v_1 + x_1v_2 + x_4v_5^2 \end{aligned} \tag{4.4}$$

*For $x = (1,1,0,0)^\intercal$, $u_1(x,\boldsymbol{v}) = 2\boldsymbol{v}_1 + \boldsymbol{v}_2$ and $u_2(x,\boldsymbol{v}) = \boldsymbol{v}_1 + \boldsymbol{v}_2$ are dependent, because they both depend on $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. For $x = (0,1,0,1)^\intercal$, $u_1(x,\boldsymbol{v}) = \boldsymbol{v}_2$ and $u_2(x,\boldsymbol{v}) = \boldsymbol{v}_1 + \boldsymbol{v}_5^2$ are independent, because $\boldsymbol{v}_2$ is independent of $\boldsymbol{v}_1$ and $\boldsymbol{v}_5^2$.*

It is possible that some, say K, of the elements of $\boldsymbol{v}$ are independent of $u_s(x, \boldsymbol{v})$ for all $s \in \mathbb{S}$. In Example (4.4), the uncertain properties are independent of $v_4$ and $v_5$ if $x = (1,1,0,0)^\intercal$ and the Happy set can be presented in lower dimensional form as

$$\mathbb{H}(x) = \left\{ v \in \mathbb{R}^2 \,\middle|\, \begin{array}{l} \mathrm{L}_1 \leq 2v_1 + v_2 \leq \mathrm{H}_1 \\ \mathrm{L}_2 \leq v_1 + v_2 \leq \mathrm{H}_2 \end{array} \right\}$$

In such case, a lower dimensional definition of the uncertain object properties and the Happy set can be given directly, by removing the irrelevant elements of $\boldsymbol{v}$.

**Corollary 4.1** *Let* $\mathrm{K}$ *elements of* $\boldsymbol{v}$ *be independent of all elements of* $u(x,\boldsymbol{v})$. *Then the uncertain object properties and corresponding Happy set can be redefined to* $u:\mathbb{R}^{\mathrm{I}}\times\mathbb{R}^{\mathrm{N-K}}\longrightarrow\mathbb{R}^{\mathrm{S}}$ *and* $\mathbb{H}(x)=\left\{v\in\mathbb{R}^{\mathrm{N-K}}|\mathrm{L}\leq u(x,v)\leq\mathrm{H}\right\}$, *where each element of* $\boldsymbol{v}$ *is dependent with at least one element of* $u(x,\boldsymbol{v})$.

A way to determine if $u_s(x,\boldsymbol{v})$ and $\boldsymbol{v}_n$ are independent, is to study the partial derivative. If for certain $x$, $\frac{\partial u_s(x,v)}{\partial v_n}=0$ for all $v\in\mathbb{V}$, then $u_s(x,\boldsymbol{v})$ and $\boldsymbol{v}_n$ are independent. For instance, the first uncertain object property of (4.4) is independent of $v_1$ for $x=(0,1,1,1)^\intercal$, since $\frac{\partial u_1(x,v)}{\partial v_1}=\frac{\partial 0v_1+v_2+v_3v_4}{\partial v_1}=0$. Similarly, one can check if $u_s(x,v)$ and $u_t(x,v)$ have any element $v_n$ in common. For the general case, if

$$\sum_{n=1}^{\mathrm{N}}\left(\max_{v\in\mathbb{V}}\left|\frac{\partial u_s(x,v)}{\partial v_n}\right|\max_{v\in\mathbb{V}}\left|\frac{\partial u_t(x,v)}{\partial v_n}\right|\right)=0 \tag{4.5}$$

then $u_s(x,\boldsymbol{v})$ and $u_t(x,\boldsymbol{v})$ are independent. This approach can be used to define the partitioning of the set of uncertain object properties as follows:

**Definition 4.1** *The partition-sets of the set of uncertain object properties are defined as* $\mathbb{G}_g(x)\subseteq\mathbb{S}$ *for* $g=1,..,\mathrm{G}$ *for which it holds that:*

$$\begin{aligned}
&\bigcup_{g=1}^{\mathrm{G}}\mathbb{G}_g(x)=\mathbb{S}\\
\forall g\neq h:\ &\mathbb{G}_g(x)\bigcap\mathbb{G}_h(x)=\emptyset\\
\forall s\in\mathbb{G}_g(x),\ t\in\mathbb{G}_h(x),\ g\neq h:\ &\sum_{n=1}^{\mathrm{N}}\left(\max_{v\in\mathbb{V}}\left|\frac{\partial u_s(x,v)}{\partial v_n}\right|\max_{v\in\mathbb{V}}\left|\frac{\partial u_t(x,v)}{\partial v_n}\right|\right)=0\\
\forall g\in\{1,..,\mathrm{G}\},s\in\mathbb{G}_g(x),\ \exists t\in\mathbb{G}_g(x):\ &\sum_{n=1}^{\mathrm{N}}\left(\max_{v\in\mathbb{V}}\left|\frac{\partial u_s(x,v)}{\partial v_n}\right|\max_{v\in\mathbb{V}}\left|\frac{\partial u_t(x,v)}{\partial v_n}\right|\right)>0
\end{aligned} \tag{4.6}$$

The last condition in Definition 4.1 makes that the (maximum) number of partitions G is unique, because this condition guarantees that each $\mathbb{G}_g(x)$, $g=1,..,\mathrm{G}$ cannot be partitioned further. It can be shown that the partitioning requires a polynomial number of steps[1].

[1] From a Graph Theory perspective, consider $\mathbb{S}$ as the set of nodes and between any two nodes for which the test (4.5) does not hold there is an arc (i.e. possible dependence). It can be shown that finding all the arcs can be done in at most $\frac{1}{2}\mathrm{S}(\mathrm{S}-1)$ steps. For each node, finding all connected nodes is a polynomial time problem (as follows from Matousek and Nesetril, 1998). Consequently, the partitions can be found in polynomial time.

**Example 4.5** *Let $v \in \mathbb{R}^4$ and the uncertain object properties be:*

$$\begin{aligned} u_1(x,v) &= 2x_1v_1 + x_2v_3 \\ u_2(x,v) &= x_2v_1 + x_1v_3 \\ u_3(x,v) &= x_3\frac{v_4}{v_2} \end{aligned}$$

*For $x = (1,1,1)^\intercal$, $u_1(x,\boldsymbol{v})$ and $u_2(x,\boldsymbol{v})$ are dependent, but they are both independent of $u_3(x,\boldsymbol{v})$. Consequently, the partition of $\mathbb{S} = \{1,2,3\}$ is $\mathbb{G}_1(x) = \{1,2\}$, $\mathbb{G}_2(x) = \{3\}$ with* $\mathrm{G} = 2$.

The partitioning of $\mathbb{S}$ corresponds to a partitioning of $\{1,..,\mathrm{N}\}$, reflecting the mutual dependence between elements of $\boldsymbol{v}$ and elements of $u(x,\boldsymbol{v})$. This partitioning is relevant in order to give an alternative definition for $R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$ based on G lower dimensional Happy sets, as shown in Corollary 4.4. The G partitions of $\{1,..,\mathrm{N}\}$ in connection to the G partitions of $\mathbb{S}$ are:

$$\mathbb{I}_g(x) = \left\{ n \in \{1,..,\mathrm{N}\} \,\middle|\, \exists_{s\in\mathbb{G}_g(x)} : \max_v \left|\frac{\partial u_s(x,v)}{\partial v_n}\right| > 0 \right\}$$

for $g = 1,..,$G. This means that for each $n \in \mathbb{I}_g(x)$, there is at least one $s \in \mathbb{G}_g(x)$ for which $\boldsymbol{v}_n$ and $u_s(x,\boldsymbol{v})$ are dependent. The number of elements in these sets is defined as $\mathrm{M}_g = |\mathbb{I}_g(x)|$ and from Corollary 4.1 follows that $\sum\limits_{g=1}^{\mathrm{G}} \mathrm{M}_g = \mathrm{N}$. In Example 4.5, the index sets are $\mathbb{I}_1(x) = \{1,3\}$ and $\mathbb{I}_2(x) = \{2,4\}$ with $\mathrm{M}_1 = 2$ and $\mathrm{M}_2 = 2$.

The concepts in Definition 4.2 and Corollary 4.2 are given, to define the decomposed Happy set as in Definition 4.3 and to show the effectiveness of Decomposition in the remainder of this section. We start with identifying which elements of $v$ are relevant in each partition:

**Definition 4.2** *Let $f_g : \mathbb{I}_g(x) \longrightarrow \{1,..,\mathrm{M}_g\}$ be a bijective function for $g = 1,..,$G. The vector $v^{[g]} \in \mathbb{R}^{\mathrm{M}_g}$ is defined by $v^{[g]}_{f_g(n)} = v_n$ for all $n \in \mathbb{I}_g(x)$*

**Corollary 4.2** *The relation between vector $v$ and vector $v^{[g]}$ is defined by an* N*-by-*$\mathrm{M}_g$ *matrix $P^{[g]}$ with $g = 1,..,$G and $P^{[g]}_{f_g(n),n} = 1$, $P^{[g]}_{f_g(n),i} = 0$ for all $n \in \mathbb{I}_g(x)$, $i \neq n$. Consequently, $v = \sum\limits_{g=1}^{\mathrm{G}} P^{[g]}v^{[g]}$ and $v^{[g]} = P^{[g]^\intercal}v$*

Hence, each column in matrix $P^{[g]}$ has one element with the value 1 and all other elements are 0 and each row has at most one element with the value 1 and all other elements 0.

**Example 4.6** *The two lower dimensional vectors corresponding the uncertain properties in Example 4.5, are $v^{[1]} = (v_1,v_3)^\intercal$ and $v^{[2]} = (v_2,v_4)^\intercal$ and the two projection matrices are*

$$P^{[1]} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \quad P^{[2]} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Since $u_s(x,v)$ with $s \in \mathbb{G}_g(x)$ is insensitive for $v_n$ with $n \notin \mathbb{I}_g(x)$ it follows that

**Corollary 4.3**

$$u_s(x,v) = u_s(x, P^{[g]}v^{[g]}) \textit{ for } s \in \mathbb{G}_g(x) \tag{4.7}$$

Definitions 4.1 and 4.2 form the basis of the definition of the lower dimensional Happy sets:

**Definition 4.3**

$$\mathbb{H}_g(x) = \left\{ z \in \mathbb{R}^{\mathrm{M}_g} \,\middle|\, \mathrm{L}_s \le u_s(x, P^{[g]}z) \le \mathrm{H}_s,\ \forall s \in \mathbb{G}_g(x) \right\} \tag{4.8}$$

From independency between any two uncertain object properties belonging to different partition sets follows immediately that the Robustness can be computed with

**Corollary 4.4**

$$R(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}(x)\right\} = \prod_{g=1}^{\mathrm{G}} \Pr\left\{\boldsymbol{v}^{(g)} \in \mathbb{H}_g(x)\right\} \tag{4.9}$$

### Happy set Decomposition effect on Bounding methods

The reason to construct a decomposition, is to reformulate the Happy set into G lower dimensional Happy sets, with the intention to estimate $R(x)$ more effectively. In particular, the effectiveness of a bound for $R(x)$ increases if the absolute difference between the bound and $R(x)$ becomes smaller. In (3.97), $\overline{\mathbb{H}}^{[\infty]}(x)$, $\overline{\mathbb{H}}^{[2]}(x)$ and $\overline{\mathbb{H}}^{[1]}(x)$ are defined as respectively the smallest cube, ball and diamond shapes that enclose the Happy set $\mathbb{H}(x)$. Let $\overline{\mathbb{H}}^{[p]}(x)$, with $p \in \{1,2,\infty\}$ be the corresponding sets that enclose the Happy set, where the maximum norm of vectors in $\overline{\mathbb{H}}^{[p]}(x)$ is $\overline{r}^{[p]}(x)$, which is defined as the *radius* of $\overline{\mathbb{H}}^{[p]}(x)$ in (3.98). By definition $\Pr_{\boldsymbol{v}}\left(\mathbb{H}(x)\right) \le \Pr_{\boldsymbol{v}}\left(\overline{\mathbb{H}}^{[p]}(x)\right)$ and $\Pr_{\boldsymbol{v}^{[g]}}\left(\mathbb{H}_g(x)\right) \le \Pr_{\boldsymbol{v}^{[g]}}\left(\overline{\mathbb{H}}_g^{[p]}(x)\right)$ for $g = 1,..,$G and consequently $\Pr_{\boldsymbol{v}}\left(\mathbb{H}(x)\right) \le \prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}}\left(\overline{\mathbb{H}}_g^{[p]}(x)\right)$. Decomposition only leads to sharper or equal upper bounds, if $\prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}}\left(\overline{\mathbb{H}}_g^{[p]}(x)\right) \le \Pr_{\boldsymbol{v}}\left(\overline{\mathbb{H}}^{[p]}(x)\right)$. Theorem 4.1 and Corollary 4.5 show that this inequality is always valid.

**Theorem 4.1**

$$\prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}}\left(\overline{\mathbb{H}}_g^{[p]}(x)\right) \le \Pr_{\boldsymbol{v}}\left(\overline{\mathbb{H}}^{[p]}(x)\right) \tag{4.10}$$

*for each norm type* $p \in \{1,2,\infty\}$

**Corollary 4.5** *If* $\exists g \in \{1,..,\mathrm{G}\}$ *with* $\overline{r}_g^{[p]}(x) < \overline{r}^{[p]}(x)$ *then*

$$\prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}} \left(\overline{\mathbb{H}}_g^{[p])}(x)\right) < \Pr_{\boldsymbol{v}} \left(\overline{\mathbb{H}}^{[p]}(x)\right) \tag{4.11}$$

*for each norm type* $p \in \{1, 2, \infty\}$

Proof of Theorem 4.1 and Corollary 4.5 is given in Appendix A.9.

The same idea holds for the lower bounds. Inscribed spheres $\underline{\mathbb{H}}^{[\infty]}(x)$, $\underline{\mathbb{H}}^{[2]}(x)$ and $\underline{\mathbb{H}}^{[1]}(x)$ are defined in (3.99) as respectively the largest cube, ball and diamond shapes enclosed by the Happy set $\mathbb{H}(x)$. The corresponding maximum norm of vectors in $\underline{\mathbb{H}}^{[p]}(x)$ is $\underline{r}^{[p]}(x)$ which is defined as the *radius* of $\underline{\mathbb{H}}^{[p]}(x)$ in (3.100) with $p \in \{1, 2, \infty\}$. Decomposition is effective for lower bounds because:

**Theorem 4.2**

$$\prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}} \left(\underline{\mathbb{H}}_g^{[p]}(x)\right) \geq \Pr_{\boldsymbol{v}} \left(\underline{\mathbb{H}}^{[p]}(x)\right) \tag{4.12}$$

*for each type* $p \in \{1, 2, \infty\}$

**Corollary 4.6** *If* $\exists g \in \{1,..,\mathrm{G}\}$ *with* $\overline{r}_g^{[p]}(x) > \overline{r}^{[p]}(x)$ *then*

$$\prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}} \left(\underline{\mathbb{H}}_g^{[p]}(x)\right) > \Pr_{\boldsymbol{v}} \left(\underline{\mathbb{H}}^{[p]}(x)\right) \tag{4.13}$$

*for each norm type* $p \in \{1, 2, \infty\}$

Proof of Theorem 4.2 and Corollary 4.6 is given in Appendix A.9.

### Happy set Decomposition effect on Sampling methods

In contrast to the exactness of the effect of Decomposition on Robustness Bounding methods, it is an open question whether or not Decomposition reduces the standard error of a Robustness sampling estimator. The empirical studies of Chapter 5, support this hypothesis. In the quest for finding a proof for this hypothesis, the following properties were discovered.

Let $\rho = \Pr_{\boldsymbol{v}} (\mathbb{H}(x)) = \Pr_{\boldsymbol{v}^{[1]}} (\mathbb{H}_1(x)) \Pr_{\boldsymbol{v}^{[2]}} (\mathbb{H}_2(x)) = \rho_1 \rho_2$, where the two Happy sets $\mathbb{H}_1(x)$ and $\mathbb{H}_2(x)$ are a decomposition of the Happy set $\mathbb{H}(x)$. The idea is that if such decomposition in two parts leads to a reduction of the standard error ($se(\cdot)$) of the Robustness estimate, then it can be proved by induction that any decomposition will lead to a standard error improvement.

**Lemma 4.1** *Let* $\hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\rho}}_1$ *and* $\hat{\boldsymbol{\rho}}_2$ *be the estimator of respectively* $\rho, \rho_1$ *and* $\rho_2$*, with standard errors* $se(\hat{\boldsymbol{\rho}}) = \sqrt{var\left[\hat{\boldsymbol{\rho}}\right]}$, $se(\hat{\boldsymbol{\rho}}_1) = \sqrt{var\left[\hat{\boldsymbol{\rho}}_1\right]}$ *and* $se(\hat{\boldsymbol{r}}_2) = \sqrt{var\left[\hat{\boldsymbol{r}}_2\right]}$. *If*

$$E\left(\hat{\boldsymbol{\rho}}_1^2\right) E\left(\hat{\boldsymbol{\rho}}_2^2\right) < E\left(\hat{\boldsymbol{\rho}}^2\right) \tag{4.14}$$

*then*

$$se(\hat{\boldsymbol{\rho}}_1 \hat{\boldsymbol{\rho}}_2) < se(\hat{\boldsymbol{\rho}})$$

**Proof.** The result follows from:

$$\begin{array}{llll} VAR(\hat{\boldsymbol{\rho}}) & = & E\left(\hat{\boldsymbol{\rho}}^2\right) & -E\left(\hat{\boldsymbol{\rho}}\right)^2 \\ VAR(\hat{\boldsymbol{\rho}}_1\hat{\boldsymbol{\rho}}_2) & = E\left(\hat{\boldsymbol{\rho}}_1^2\hat{\boldsymbol{\rho}}_2^2\right) - E\left(\hat{\boldsymbol{\rho}}_1\hat{\boldsymbol{\rho}}_2\right)^2 = & E\left(\hat{\boldsymbol{\rho}}_1^2\right) E\left(\hat{\boldsymbol{\rho}}_2^2\right) & -E\left(\hat{\boldsymbol{\rho}}\right)^2 \end{array}$$

■

**Corollary 4.7** *Let $\overset{\text{mc}}{\boldsymbol{\rho}}$, $\overset{\text{mc}}{\boldsymbol{\rho}}_1$ and $\overset{\text{mc}}{\boldsymbol{\rho}}_2$ be Monte Carlo estimators of respectively $\rho$, $\rho_1$ and $\rho_2$, then*

$$se(\overset{\text{mc}}{\boldsymbol{\rho}}) = se(\overset{\text{mc}}{\boldsymbol{\rho}}_1\overset{\text{mc}}{\boldsymbol{\rho}}_2)$$

**Proof.** For Monte Carlo estimation

$$\begin{array}{ll} VAR(\overset{mc}{\boldsymbol{\rho}}) = \rho - \rho^2 & E(\overset{mc}{\boldsymbol{\rho}}^2) = \rho \\ VAR(\overset{mc}{\boldsymbol{\rho}}_1) = \rho_1 - \rho_1^2 & E(\overset{mc}{\boldsymbol{\rho}}_1^2) = \rho_1 \\ VAR(\overset{mc}{\boldsymbol{\rho}}_2) = \rho_2 - \rho_2^2 & E(\overset{mc}{\boldsymbol{\rho}}_2^2) = \rho_2 \end{array} \tag{4.15}$$

Consequently

$$E\left(\overset{mc}{\boldsymbol{\rho}}_1^2\right) E\left(\overset{mc}{\boldsymbol{\rho}}_2^2\right) = \rho_1\rho_2 = \rho = E\left(\overset{mc}{\boldsymbol{\rho}}^2\right)$$

■

The conclusion is that Decomposition does not lead to an efficiency improvement of the MC method. However, the case studies in Chapter 5 show that Decomposition leads to an efficiency improvement for most of the other Robustness estimation methods based on sampling.

### 4.3.2 Happy set Compression

The possibility of Happy set Compression depends on both the uncertain restrictions as well as the type of density function of the stochastic vector $\boldsymbol{v}$. The underlying principle of the compression method is the following. The Happy set can be compressed for constant $x$, if there exists an alternative expression for all uncertain object properties $s = 1,..,$S

$$u_s(x, \boldsymbol{v}) = w_s(x, \boldsymbol{a}) \tag{4.16}$$

where random vector $\boldsymbol{a}$ has A < N elements.

**Example 4.7** *Examples of such expression for $\boldsymbol{v}_i \sim N(0,1)$ are*

$$\begin{array}{llll} u_1([x_1, x_2]^\intercal, [\boldsymbol{v}_1, \boldsymbol{v}_2]^\intercal) & = x_1\boldsymbol{v}_1 + x_2\boldsymbol{v}_2 & = \sqrt{x_1^2 + x_2^2}\boldsymbol{a}_1 & = w_1(x, \boldsymbol{a}_1) \\ u_2([x_1], [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]^\intercal) & = \frac{x_1\boldsymbol{v}_1}{\sqrt{\boldsymbol{v}_2^2 + \boldsymbol{v}_3^2}} & = \frac{x_1}{\sqrt{2}}\boldsymbol{a}_2 & = w_2(x, \boldsymbol{a}_2) \end{array}$$

*where $\boldsymbol{a}_1 \sim N(0,1)$ and $\boldsymbol{a}_2 \sim \boldsymbol{t}(2)$ is a random variable following the student t-distribution with 2 degrees of freedom.*

The topic of the next sections is Robustness computation for Happy sets, where $u_s(x, v)$ is a linear function of $v$ for constant $x$. In such case, compression is particularly relevant if $\boldsymbol{v}_n \sim N(0,1)$ for $n = 1,..,$N.

## 4.4 Polyhedral Happy set

The uncertain properties $u_s(x, v)$, of the cases discussed in 5.2 and 5.4, are linear functions of $v$, for a fixed $x$. In such situation, the Happy set is a polyhedral set. Characteristic for polyhedral Happy sets is that they are defined by uncertain object properties with so-called *linear uncertainty*, i.e. a function linear in $v$:

$$u_s(x, v) = f_s(x)^\intercal (Tv + \mu) + g_s(x) \text{ for all } s = 1, .., \mathrm{S} \tag{4.17}$$

with continuous vector function $f_s : \mathbb{R}^\mathrm{I} \to \mathbb{R}^\mathrm{N}$ and scalar function $g_s : \mathbb{R}^\mathrm{I} \to \mathbb{R}$ for all $s = 1, ..,$S. The N×N matrix $T$ typically represents covariance information $TT^\intercal = \Sigma$ of the model. In vector notation, the uncertain object properties read

$$u(x, v) = f(x)^\intercal (Tv + \mu) + g(x) = f(x)^\intercal Tv + f(x)^\intercal \mu + g(x) \tag{4.18}$$

with

$$f : \mathbb{R}^\mathrm{I} \longrightarrow \mathbb{R}^{\mathrm{N}\times\mathrm{S}}$$
$$g : \mathbb{R}^\mathrm{I} \longrightarrow \mathbb{R}^\mathrm{S}$$

and

$$f(x) = (f_1(x), .., f_s(x), ..f_S(x))$$
$$g(x) = (g_1(x), .., g_s(x), ..g_S(x))^\intercal$$

where the range of $f(x)$ is an N×S matrix. In particular, linear regression models can be defined by notation (4.18) as an inner product of possible realisations of uncertain regression estimators $\boldsymbol{\beta} = (T\boldsymbol{v} + \mu)$ and some vector valued functions $f_s(x)$. The application of this type of model is discussed in Sections 5.2 and 5.4.

Polyhedral Happy sets have specific properties which can be exploited for Robustness Estimation. In this section, the benefit of *dimension reduction* in the context of Robustness bounding methods and Robustness sampling methods is discussed.

### 4.4.1 Polyhedral Happy set Dimension Reduction

The polyhedral structure of the Happy set can be exploited for the two introduced types of Dimension Reduction:

**Polyhedral Happy set Decomposition**

Let $A(x) = f(x)^\intercal T$, then

$$u(x, v) = A(x)v + f(x)^\intercal \mu + g_s(x) \tag{4.19}$$

The independency criterion (4.5) introduced in Section 4.3.1 in the situation of a Polyhedral Happy set, can be expressed as:

$$\sum_{n=1}^{\mathrm{N}} \left( \max_{v \in \mathbb{V}} \left| \frac{\partial u_s(x, v)}{\partial v_n} \right| \max_{v \in \mathbb{V}} \left| \frac{\partial u_t(x, v)}{\partial v_n} \right| \right) = \sum_{n=1}^{\mathrm{N}} |A_{s,n}(x) A_{t,n}(x)| = 0 \tag{4.20}$$

for any two uncertain object properties $u_s(x, v)$ and $u_t(x, v)$ for $s \neq t$.

### Polyhedral Happy set Compression

In the situation of $N > S$ (i.e. more random variables than uncertain object properties) and for fixed $x$, the function $u : \mathbb{R}^{\mathrm{I}} \times \mathbb{R}^{\mathrm{N}} \longrightarrow \mathbb{R}^{\mathrm{S}}$ maps a high dimensional space onto a lower dimensional space. The random vector $\boldsymbol{v}$ with N elements, defines a random vector with S elements via $u(x, v)$. Let

$$\boldsymbol{y}_s(x) = u_s(x, \boldsymbol{v}) \tag{4.21}$$

then, without loss of generality

$$\begin{aligned}
E(\boldsymbol{y}_s(x)) &= f_s(x)^\intercal \mu + g_s(x) \\
VAR(\boldsymbol{y}_s(x)) &= f_s(x)^\intercal TT^\intercal f_s(x) \\
COV(\boldsymbol{y}_s(x), \boldsymbol{y}_t(x)) &= \tfrac{1}{2}\left(VAR\left(\boldsymbol{y}_s(x) + \boldsymbol{y}_t(x)\right) - VAR(\boldsymbol{y}_s(x)) - VAR(\boldsymbol{y}_t(x))\right) \\
&= \tfrac{1}{2}\left(f_s(x)^\intercal TT^\intercal f_t(x) + f_t(x)^\intercal TT^\intercal f_s(x)\right) \\
&= f_s(x)^\intercal TT^\intercal f_t(x) \text{ (the latter since } TT^\intercal \text{ is symmetric)}
\end{aligned} \tag{4.22}$$

with $s = 1, .., \mathrm{S}$ and $t = 1, .., \mathrm{S}$. The covariance matrix $C(x) \in \mathbb{R}^{\mathrm{S}\times\mathrm{S}}$ of random vector $\boldsymbol{y}(x)$ equals:

$$C(x) = f(x)^\intercal TT^\intercal f(x) \tag{4.23}$$

From this definition follows that $C(x)$ is positive-semidefinite or positive-definite. Let $\boldsymbol{z}$ be a random vector with realisations $z \in \mathbb{R}^{\mathrm{S}}$, $VAR(\boldsymbol{z}_s) = 1$, $E(\boldsymbol{z}_s) = 0$ and $COV(\boldsymbol{z}_s, \boldsymbol{z}_t) = 0$, for $s = 1, ..$S, $t = 1, ..$S and $s \neq t$. The correlated elements of the random vector $\boldsymbol{y}(x)$ can be transformed to central and uncorrelated elements, with

$$\boldsymbol{y}(x) = u(x, \boldsymbol{v}) = L(x)\boldsymbol{z} + f(x)^\intercal \mu + g(x) \tag{4.24}$$

where $L(x)$ is the S×S lower triangular matrix decomposition of $C(x)$, such that

$$C(x) = L(x)L^\intercal(x) \tag{4.25}$$

The Happy set in the $z$-space can be defined as

$$^{[z]}\mathbb{H}(x) = \left\{z \in \mathbb{R}^{\mathrm{S}} | \mathrm{L} \leq L(x)z + f(x)^\intercal \mu + g(x) \leq \mathrm{H}\right\} \tag{4.26}$$

If $C(x)$ is positive-definite, then $L(x)$ can be obtained via Cholesky decomposition. If $C(x)$ is positive-semidefinite, then $L(x)$ can be obtained via QR-decomposition[2] with $T^\intercal f(x) = Q(x)L^\intercal(x)$, with $Q^\intercal(x)Q(x) = I$ and lower triangular $L(x)$:

$$f(x)^\intercal TT^\intercal f(x) = \left(T^\intercal f(x)\right)^\intercal T^\intercal f(x) = (Q(x)L^\intercal(x))^\intercal Q(x)L^\intercal(x) = L(x)L^\intercal(x)$$

Compression of a polyhedral Happy set, is particularly useful in the situation of $\boldsymbol{v}_n \sim N(0, 1)$ for all $n = 1, ..$N, because then $\boldsymbol{z}_s \sim N(0, 1)$ for all $s = 1, ..$S. For the general case, compression of polyhedral Happy set is useful in the bounding context, because this leads to more effective bounds as shown in the following section.

[2] In linear algebra textbooks (e.g. Golub and Van Loan (1989, page 211)), QR-decomposition of a real S×N (with S≤N) matrix $A$ is defined as a decomposition such that $A = QR$, where $Q$ is an S×N orthogonal matrix (meaning that $Q'Q = I$, with $I$ the identity matrix) and $R$ is an S×S upper triangular matrix. In the notation of this thesis, $R(x)$ is already used for Robustness. The matrix $L$ is introduced to prevent ambiguous notation: In this thesis, the notation "$L$" is equivalent to the linear algebra textbooks notation "$R$", referring to the upper triangular matrix in QR-decomposition.

### 4.4.2 Robustness Bounding method for polyhedral Happy set

The Robustness bounding methods are introduced in Section 3.8. The Robustness bounding methods use sets that inscribe the Happy set or are enclosed by the Happy. The inscribed or enclosing sets have such geometric shapes, that it is relatively easy to compute their probability mass in comparison to computing the probability mass of the Happy set. In the following, $r$ is the radius of such inscribed or enclosing sets. For polyhedral Happy sets, the most effective radius can be found as follows.

**Lower bounds**

Theorem A.4 in Appendix A.10 gives an easy expression for the minimum distance of the origin to the plane $a^\intercal v = b$, where $a \in \mathbb{R}^{\mathrm{N}}$ and $b \in \mathbb{R}$:

$$\min_{v\in\mathbb{R}^{\mathrm{N}}} \left\{ \|v\|_p \,\middle|\, a^\intercal v = b \right\} = \frac{|b|}{\|a\|_{\frac{p}{p-1}}}$$

This expression can be used for solving the optimal radius problems in (3.100). For polyhedral Happy sets, the radius of the largest diamond $\underline{\mathbb{H}}^{[1]}(x)$, ball $\underline{\mathbb{H}}^{[2]}(x)$ and cube $\underline{\mathbb{H}}^{[\infty]}(x)$ inside such Happy set can be solved analytically with:

$$\underline{r}^{[1]}(x) = \min_{s} \min_{\mathrm{B}=\mathrm{L}_s,\mathrm{H}_s} \left\{ \frac{|\mu^\intercal f_s(x) + g_s(x) - \mathrm{B}|}{\|f_s(x)'T\|_\infty} \right\} \tag{4.27}$$

$$\underline{r}^{[2]}(x) = \min_{s} \min_{\mathrm{B}=\mathrm{L}_s,\mathrm{H}_s} \left\{ \frac{|\mu^\intercal f_s(x) + g_s(x) - \mathrm{B}|}{\|f_s(x)'T\|_2} \right\} \tag{4.28}$$

$$\underline{r}^{[\infty]}(x) = \min_{s} \min_{\mathrm{B}=\mathrm{L}_s,\mathrm{H}_s} \left\{ \frac{|\mu^\intercal f_s(x) + g_s(x) - \mathrm{B}|}{\|f_s(x)'T\|_1} \right\} \tag{4.29}$$

Similarly, the analytical expressions for the largest radii of the enclosed diamond ${}^{[z]}\underline{\mathbb{H}}^{[1]}(x)$, ball ${}^{[z]}\underline{\mathbb{H}}^{[2]}(x)$ and cube ${}^{[z]}\underline{\mathbb{H}}^{[\infty]}(x)$ are

$${}^{[z]}\underline{r}^{[1]}(x) = \min_{s} \min_{\mathrm{B}=\mathrm{L}_s,\mathrm{H}_s} \left\{ \frac{|\mu^\intercal f_s(x) + g_s(x) - \mathrm{B}|}{\|l_s(x)\|_\infty} \right\} \tag{4.30}$$

$${}^{[z]}\underline{r}^{[2]}(x) = \min_{s} \min_{\mathrm{B}=\mathrm{L}_s,\mathrm{H}_s} \left\{ \frac{|\mu^\intercal f_s(x) + g_s(x) - \mathrm{B}|}{\|l_s(x)\|_2} \right\} \tag{4.31}$$

$${}^{[z]}\underline{r}^{[\infty]}(x) = \min_{s} \min_{\mathrm{B}=\mathrm{L}_s,\mathrm{H}_s} \left\{ \frac{|\mu^\intercal f_s(x) + g_s(x) - \mathrm{B}|}{\|l_s(x)\|_1} \right\} \tag{4.32}$$

where $l_s(x)$ is the s-th row of $L(x)$: $L(x)^\intercal = [l_1(x), .., l_s(x), .., l_S(x)]$.

We now discuss Theorem 4.3 which is in particular relevant for practically solving the largest radius problem for $p = 2$. Consider the situation that there are less uncertain restrictions than the number of elements of vector $\boldsymbol{v}$, i.e. S<N. Theorem 4.3 tells us that the radius of the largest enclosed ball in the high dimensional Happy set $\mathbb{H}(x) \subseteq \mathbb{R}^{\mathrm{N}}$ is *identical* to the radius of the largest enclosed ball in the lower dimensional Happy set ${}^{[z]}\mathbb{H}(x) \subseteq \mathbb{R}^{\mathrm{S}}$. This means that the radius can be computed conveniently in the high dimensional Happy set, thus getting around the computationally expensive Cholesky or QR-factorization and use the results to make conclusions about the lower bound in the lower dimensional Happy set.

**Theorem 4.3** *For fixed $x$,*

$$ {}^{[z]}\underline{r}^{[2]}(x) = \underline{r}^{[2]}(x) $$

**Proof.** The covariance matrix of $\boldsymbol{y}(x)$ is

$$ f(x)^{\intercal} T T^{\intercal} f(x) = C(x) = L(x) L(x)^{\intercal} $$

The $s$-th diagonal element of the covariance matrix of $\boldsymbol{y}(x)$ is

$$ f_s(x)' T T' f_s(x) = C_{s,s}(x) = l_s(x)^{\intercal} l_s(x) $$

which implies that

$$ \left\| f_s(x)^{\intercal} T \right\|_2 = \sqrt{f_s(x)^{\intercal} T T^{\intercal} f_s(x)} = \sqrt{l_s(x)^{\intercal} l_s(x)} = \left\| l_s(x) \right\|_2 $$

and that (4.28) and (4.31) are identical. ■

### Upper bounds

In the situation of a polyhedral Happy set, the optimisation problems related to finding the smallest diamond $\overline{\mathbb{H}}^{[1]}(x)$, ball $\overline{\mathbb{H}}^{[2]}(x)$ and cube $\overline{\mathbb{H}}^{[\infty]}(x)$ enclosing the Happy set, as defined in (3.98), read:

$$ \begin{aligned} \overline{r}^{[1]}(x) &= \max_{v \in \mathbb{H}(x)} \|v\|_1 &&= \max_{v \in \mathbb{H}(x)} \sum_{n=1}^{\mathrm{N}} |v_n| \\ \overline{r}^{[2]}(x) &= \max_{v \in \mathbb{H}(x)} \|v\|_2 &&= \max_{v \in \mathbb{H}(x)} \sqrt{v'v} \\ \overline{r}^{[\infty]}(x) &= \max_{v \in \mathbb{H}(x)} \|v\|_\infty &&= \max_{v \in \mathbb{H}(x)} \max_{1 \le n \le N} |v_n| \end{aligned} \tag{4.33} $$

The first two ($\overline{r}^{[1]}(x)$ and $\overline{r}^{[2]}(x)$) are found by *maximising* a convex function over a convex set. In (Bazaraa et al., 1993) it is explained that such problems can be NP-hard. The optimisation problem for finding $\overline{r}^{[\infty]}(x)$ can be found as follows:

$$ \begin{aligned} \overline{r}^{[\infty]}(x) &= \max_{v \in \mathbb{H}(x)} \|v\|_\infty \\ &= \max_{v \in \mathbb{H}(x)} \max_{1 \le n \le \mathrm{N}} |v_n| \\ &= \max \left\{ \overline{r}_1^{[\infty]}, .., \overline{r}_n^{[\infty]}, .., \overline{r}_{\mathrm{N}}^{[\infty]} \right\} \end{aligned} $$

with

$$ \overline{r}_n^{[\infty]}(x) = \max \left\{ \max_{v \in \mathbb{H}(x)} (v_n), \max_{v \in \mathbb{H}(x)} (-v_n) \right\} $$

The conclusion is that Robustness lower bounds can be computed analytically for polyhedral Happy sets. A Robustness upper bound for a polyhedral Happy corresponds to $\overline{r}^{[\infty]}(x)$ and can be found in polynomial time, since such problem consists of solving 2N Linear Programming problems.

## 4.5 Warm Start Robustness optimisation approach

The Matlab® optimisation software (FMINCON) is used to solve the RP problems described in the the cases of Chapter 5. Solving (2.8) for the case of Section 5.2 with the FMINCON solver, in general requires a starting point to start the optimisation iterations. For that purpose, L starting points $x^{[l]}$ $(1,..,\mathrm{L})$ are chosen uniformly[3] distributed over $\mathbb{X}$. While working on the cases of Section 5.2, it became clear that for some random starting points, the Happy set $\mathbb{H}(x)$ may be empty in a neighbourhood $\left\{x \in \mathbb{R}^{\mathrm{I}} \middle| \, |x_i^{[l]} - x_i| \leq \Delta, \, i = 1,..,\mathrm{I}\right\}$ around the starting point $x^{[l]}$ where $\Delta$ is the Finite Differencing (FD) step size. Consequently, for such starting points, the finite differencing approach for estimating the gradient $\nabla R(x)$ based on sampling estimation methods will estimate a gradient with length zero and results in terminating the optimisation algorithm, since the First Order optimality conditions are satisfied.

To overcome this problem, the following so-called *Warm Start* (WS) method is suggested, similar to the penalty function approach (3.111) illustrated in Section 3.8. Let $\hat{R}(x)$ be any of the Robustness estimates for $R(x)$ based on sampling, i.e. an estimate based on the MC, SMC, N-1MC, DS or ES method. Let $\underline{r}^{[2]}(x)$ be the radius as defined in (3.111), which is the distance from $E(\boldsymbol{v})$ to the closed boundary of $\mathbb{H}(x)$. This radius is non-negative if $E(\boldsymbol{v}) \in \mathbb{H}(x)$ and negative if $E(\boldsymbol{v}) \notin \mathbb{H}(x)$. The Warm Start estimate: $\overset{\mathrm{ws}}{R} : \mathbb{R}^{\mathrm{N}} \longrightarrow (-\infty, 1]$ is defined as

$$\overset{\mathrm{ws}}{R}(x) = \begin{cases} \hat{R}(x) & \text{if } \hat{R}(x) > 0 \\ \underline{r}^{[2]}(x) & \text{if } \hat{R}(x) = 0 \text{ and } \underline{r}^{[2]}(x) < 0 \\ 0 & \text{elsewhere} \end{cases} \tag{4.34}$$

and can be maximised with the FMINCON solver to find the optimal Robustness estimate

$$\hat{R}^* = \max_{x \in \mathbb{X}} \left[ \overset{\mathrm{ws}}{R}(x) \right] \tag{4.35}$$

The underlying principle of the WS estimate function is that, as long as $\underline{r}^{[2]}(x) < 0$, maximising $\overset{\mathrm{ws}}{R}(x)$ is effectively the same as minimising the distance between $E(\boldsymbol{v})$ and the point in the Happy set closest to $E(\boldsymbol{v})$. Consequently, as soon as this distance becomes non-negative, then $E(\boldsymbol{v}) \in \mathbb{H}(x)$, which means that $\mathbb{H}(x)$ is not empty anymore. This idea has a parallel to the first phase of the Simplex method in Linear Programming, where the first phase results in a feasible starting point. Similarly, the first phase of the Warm Start method results in finding an $x$ for which $\hat{R}(x) > 0$.

## 4.6 Comparison of RP methods

Case studies are discussed in Chapter 5, which are solved with Robustness Programming (RP) methods. The primary purpose of the cases, is to investigate the performance of

[3] In the studied cases the set $\mathbb{X}$ is typically a polytope. Devroye (1986) discusses how random points can be generated, that are uniformly distributed over $\mathbb{X}$.

the RP methods and conclude which RP methods perform above average in a practical relevant context.

The following is an overview of all RP methods, of which the performance is investigated:

- The MC, SMC, N-1MC, DS and ES Robustness sampling estimation methods as given in Chapter 3, with M the number of samples used

- The Robustness lower[4] [5] bounding methods as given in Chapter 3 and are called the Diamond, Ball and Cube method respectively

- The Decomposition and Compression methods as given in Section 4.3; The Warm Start method as defined in (4.34)

Parameter setting q=($q_E$,$q_C$,$q_D$,$q_W$,$q_S$) determines the RP method selection. The possible parameter values for q are given in Table 4.1. For example if q= (DS, Yes, Yes, No,M=100), then the RP method selection for estimating $R(x)$ is to use the DS method based on M=100 number of samples in combination with Decomposition and Compression without the Warm Start method. Let K be the number of values that parameter vector q can have. From Table 4.1 it follows that in theory there are K=8×2×2×2×2 = 128 alternative values for q. Table 4.2 illustrates that the values $q^{[1]}$,..,$q^{[\kappa]}$,..,$q^{[K]}$ can be identified by an index number $\kappa = 1, ...,$K. The parameter $q^{[\kappa]}$ is called the RP method combination $\kappa$. For example, RP method combination $\kappa = 63$ identifies $q^{[63]}$ = (DS, Yes, Yes, Yes, M=100). In practice not all of these K combinations are used. For instance if $\boldsymbol{v}$ is normally distributed, then the ES and Diamond method do not apply. Let $\underline{K}$ be the maximum number of feasible combinations of RP methods and let $\kappa = 1, .., \underline{K}$ identify each feasible combination of RP methods. For example if we fix WS=Yes and $\boldsymbol{v}$ is normally distributed, then $\underline{K} = 6 \times 2 \times 2 \times 2 = 48$.

Table 4.1: Robustness Programming parameter settings

| Parameter | Meaning | Values |
|---|---|---|
| $q_E$ | Estimation method | MC, SMC, N-1MC, DS, ES, Diamond, Ball, Cube |
| $q_D$ | Decomposition | Yes, No |
| $q_C$ | Compression | Yes, No |
| $q_W$ | Warm Start | Yes, No |
| $q_S$ | Number of samples | M=100, M=400 |

Two computers[6], respectively called computer A and computer B, are used to have sufficient capacity to do numerical experiments in a timely way. An approach to investigate the performance of each RP method $\kappa$, in an optimisation context, is to compare the

[4] In Section 3.8 is illustrated that maximising an upper bound is not practical, since this can result in finding an unbounded Happy set, which is not necessarily associated to the maximal Robustness.

[5] These methods refer to estimating a lower bound for $R(x)$ based on the Diamond set (3.102), the Ball set (3.106) and the Cube set (3.109) where respectively $\underline{r}^{[1]}(x)$, $\underline{r}^{[2]}(x)$ and $\underline{r}^{[\infty]}(x)$ follow from (3.111).

[6] Computer A is a 1,83GHz Intel Core Duo computer with 1GB of memory and computer B is a 2GHz Intel Core Duo computer with 2GB of memory

Table 4.2: Illustration of all parameter setting values

| Index number ($\kappa$) | parameter setting ($\mathrm{q}^{[\kappa]}$) | Value |
|---|---|---|
| 1 | $\mathrm{q}^{[1]}$ | (MC, No, No, No,M=100) |
| 2 | $\mathrm{q}^{[2]}$ | (MC, No, No, No,M=400) |
| 3 | $\mathrm{q}^{[3]}$ | (MC, No, No, Yes,M=100) |
| 4 | $\mathrm{q}^{[4]}$ | (MC, No, No, Yes,M=400) |
| 5 | $\mathrm{q}^{[5]}$ | (MC, No, Yes, No,M=100) |
| 6 | $\mathrm{q}^{[6]}$ | (MC, No, Yes, No,M=400) |
| 7 | $\mathrm{q}^{[7]}$ | (MC, No, Yes, Yes,M=100 |
| 8 | $\mathrm{q}^{[8]}$ | (MC, No, Yes, Yes,M=400) |
| ⋮ | ⋮ | ⋮ |
| 57 | $\mathrm{q}^{[57]}$ | (DS, Yes, No, No,M=100) |
| 58 | $\mathrm{q}^{[58]}$ | (DS, Yes, No, No,M=400) |
| 59 | $\mathrm{q}^{[59]}$ | (DS, Yes, No, Yes,M=100) |
| 60 | $\mathrm{q}^{[60]}$ | (DS, Yes, No, Yes,M=400) |
| 61 | $\mathrm{q}^{[61]}$ | (DS, Yes, Yes, No,M=100) |
| 62 | $\mathrm{q}^{[62]}$ | (DS, Yes, Yes, No,M=400) |
| 63 | $\mathrm{q}^{[63]}$ | (DS, Yes, Yes, Yes,M=100 |
| 64 | $\mathrm{q}^{[64]}$ | (DS, Yes, Yes, Yes,M=400) |
| ⋮ | ⋮ | ⋮ |
| 127 | $\mathrm{q}^{[127]}$ | (Cube, Yes, Yes, Yes,M=100) |
| 128 | $\mathrm{q}^{[128]}$ | (Cube, Yes, Yes, Yes,M=400) |

time until the FMINCON solver converges to a (local) optimum. However, the cases that are discussed in Sections 5.2 and 5.4 appear to result in such long computation times, that it is not practical to wait for the algorithm to converge to a solution. For example, it takes approximately 2 weeks to compute 20 optimisation iterations of the optimisation problem as discussed in Section 5.2, on computer A with the Matlab FMINCON solver, for all starting points $x^{[l]}$, $l = 1, .., 120$ and RP methods $\kappa = 1, .., \underline{\mathrm{K}}$ . Experiments showed that for most $\mathrm{q}^{[\kappa]}$, the FMINCON solver requires more than 60 iterations to converge. Running 60 iterations instead of 20 iterations, implies more than 6 weeks computation time per case and is considered not practical.

Initially, the choice was made to compare the methods after a fixed amount of time and after a fixed number of iterations. However, there are two objections to base the the method comparison on the computation time. Firstly, the computing platform (e.g. computers and operating system) has to be configured in such a way that the processing speed is constant, such that all experiments are comparable. This is not the case for the available configuration of computers A and B. Secondly, the computation time depends on the quality of the software implementation. In Chapter 5, the mean computation time is given to illustrate the performance of each RP method, based the results of the case studies. Although the mean computation time of each RP method does give practical information about the current software implementation, the mean computation time does not give an objective statistic about the RP method performance. Therefore the choice was made to investigate the RP methods by comparing the Robustness of the iterate reached after a fixed number of iterations. The following two algorithms are defined for this purpose and are given in detail at the end of this section.

- Algorithm 4.1 is the pseudo-code for the Robustness estimate function $f_{\kappa}^{\mathrm{RPEst}} : \mathbb{R}^{\mathrm{N}} \longrightarrow (-\infty, 1]$, where $f_{\kappa}^{\mathrm{RPEst}}(x)$ is an estimate for $R(x)$, given $x \in \mathbb{R}^{\mathrm{N}}$ and parameter vector $\mathrm{q}^{[\kappa]}$. The negative range is due to the WS method.

- Algorithm 4.2 is the pseudo-code for $f_{\kappa}^{\mathrm{RPSolve}} : \mathbb{R}^{\mathrm{N}} \longrightarrow (-\infty, 1]$, which returns the highest Robustness value found within 20 optimisation iterations of maximising $f_{\kappa}^{\mathrm{RPEst}}(x)$ with the FMINCON solver, given parameter vector $\mathrm{q}^{[\kappa]}$ and starting point $x^{[l]}$. The output of the function is $f_{\kappa}^{\mathrm{RPSolve}}\left(x^{[l]}\right) = \tilde{R}\left(x^{[l,\kappa]*}\right)$, where $x^{[l,\kappa]*}$ is the corresponding best design that is found within 20 iterations of the FMINCON solver. The Robustness estimate $\tilde{R}(x)$ has an accuracy upper bound $\hat{se}\left(\overset{\mathrm{mc}}{\tilde{\boldsymbol{R}}}(x)\right) \leq 0.0035$ given 20000 samples[7] and is explained in Appendix A.3

**Example 4.8** *An illustration is given of possible parameter selections and outcomes. In this example, the RP problem of Section 5.2 (the mixture design case of Unilever) is solved for three given starting points $x^{[1]}$, $x^{[2]}$ and $x^{[3]}$. The following selections of RP methods are investigated:*

$$\begin{aligned}
\mathrm{q}^{[11]} &= (\text{SMC, Yes, No, Yes, M=100}) \\
\mathrm{q}^{[15]} &= (\text{SMC, Yes, Yes, Yes, M=100}) \\
\mathrm{q}^{[59]} &= (\text{DS, Yes, No, Yes, M=100}) \\
\mathrm{q}^{[63]} &= (\text{DS, Yes, Yes, Yes, M=100})
\end{aligned}$$

*The results after 20 iterations for the given RP methods are:*

$$\begin{array}{lll}
f_{11}^{RPSolve}\left(x^{[1]}\right) = 0.92 & f_{11}^{RPSolve}\left(x^{[2]}\right) = -2.13 & f_{11}^{RPSolve}\left(x^{[3]}\right) = 0.99 \\
f_{15}^{RPSolve}\left(x^{[1]}\right) = 0.99 & f_{15}^{RPSolve}\left(x^{[2]}\right) = -2.13 & f_{15}^{RPSolve}\left(x^{[3]}\right) = 0.97 \\
f_{59}^{RPSolve}\left(x^{[1]}\right) = 0.94 & f_{59}^{RPSolve}\left(x^{[2]}\right) = -2.13 & f_{59}^{RPSolve}\left(x^{[3]}\right) = 0.97 \\
f_{63}^{RPSolve}\left(x^{[1]}\right) = 0.97 & f_{63}^{RPSolve}\left(x^{[2]}\right) = -2.13 & f_{63}^{RPSolve}\left(x^{[3]}\right) = 0.98
\end{array}$$

*For example, the Robustness values found by the solver, for starting point $x^{[1]}$ and method $q^{[15]}$, is 0.99 after 20 iterations, showing that $q^{[15]}$ is the best performing method, given starting point $x^{[1]}$. For starting point $x^{[2]}$ the Robustness could not be determined and the Warm Start value is returned instead (which is identical for all methods). The method $q^{[11]}$ is the best performing method, given starting point $x^{[3]}$.*

The computation time to compute $f_{\kappa}^{\mathrm{RPSolve}}\left(x^{[l]}\right)$ is recorded on each computer, for each starting point $x^{[l]}$ and method $\kappa$. The computation times are given as an illustration in Sections 5.2, 5.3 and 5.4 and are based on measurements from the relatively faster computer B[8].

The outcome of the experiment depends on the starting point and the RP method that is used, as illustrated in Example 4.8. The average RP method performance can be investigated by repeating the experiments for different starting points. Let $\boldsymbol{S} = (\boldsymbol{x}^{[1]}, .., \boldsymbol{x}^{[\mathrm{L}]})$

[7] Due to memory limits of 1GB, it was not practical to use more than 20000 samples

[8] A software improvement was necessary to collect these statistics correctly, which implied to repeat the computations. It was not practical to repeat all computations since that would take several weeks. Therefore the measured average computation times are based on 10 starting points.

be a matrix with L random starting points and let matrix $S$ be a sample of L starting points. In the following, two performance indicators are defined. The main idea for these performance indicators is that, for a given starting point $x^{[l]}$, the non-negative values of $\tilde{R}\left(x^{[l,1]*}\right), .., \tilde{R}\left(x^{[l,\kappa]*}\right), .., \tilde{R}\left(x^{[l,\mathrm{K}]*}\right)$ are selected and ranked in descending order. The first element in such ranking and the subsequent elements having the same value, identify the best performing methods (like $\kappa = 15$ in Example 4.8 given starting point $x^{[1]}$). Finally, the percentage that RP method $\kappa$ is the best performing method, can be estimated by repeating the experiment for all L starting points, and count how often RP method combination $\kappa$ is the best performing method of the L experiments.

The outcome of this ranking approach can be, that there is more than one best performing method in the ranking, given starting point $x^{[l]}$, i.e. these best performing methods perform equally well. The following performance indicators are defined, which are consistent in the sense that they correct[9] for double counting and the sum over the compared methods add up to 100%:

- **The RP method performance indicator**
  We define $\pi_S^{[\kappa]}$ as the number of times that RP method $\mathrm{q}^{[\kappa]}$ is the best performing method, during L experiments and is expressed as a percentage of L. The performance indicator for RP method $\kappa = 1, ..,$K and starting points $S$, is defined as

$$\pi_S^{[\kappa]} = \frac{100\%}{\mathrm{L}} \sum_{l\in 1}^{\mathrm{L}} \frac{I\left(f_\kappa^{\mathrm{RPSolve}}\left(x^{[l]}\right) \geq f_\gamma^{\mathrm{RPSolve}}\left(x^{[l]}\right) \geq 0, \gamma \in \{1, .., \underline{\mathrm{K}}\}\right)}{\sum\limits_{\gamma\in 1}^{\underline{\mathrm{K}}} I\left(f_\kappa^{\mathrm{RPSolve}}\left(x^{[l]}\right) = f_\gamma^{\mathrm{RPSolve}}\left(x^{[l]}\right)\right)} \tag{4.36}$$

  where $I\left(\alpha\right)$ is an indicator function, that returns the value 1 if $\alpha$ is true and 0 if $\alpha$ is false. The starting points $x^{[l]}$ are taken from matrix $S$ and define the L experiments[10]. The nominator $I\left(f_\kappa^{\mathrm{RPSolve}}\left(x^{[l]}\right) \geq f_\gamma^{\mathrm{RPSolve}}\left(x^{[l]}\right) \geq 0, \gamma \in \{1, .., \underline{\mathrm{K}}\}\right)$ counts how often method $\kappa$ is better or equal than all the other methods $\gamma$. The denominator count the methods that perform as good as the best RP method. Notice that the nominator is zero If $\kappa$ is not one of the best methods, in which case the denominator is irrelevant. It can be shown that $\sum\limits_{\kappa\in 1}^{\underline{\mathrm{K}}} \pi_S^{[\kappa]} = 100\%$.

- **The aggregated RP method performance indicator**
  To measure the individual effect of RP methods in $\mathbb{M}$={MC, SMC, N-1MC, DS, ES, Diamond, Cube, Ball, Decom., Comp., M= 400, WS}, we define $\Pi^{[type]}$ with $type \in \mathbb{M}$ as

[9] The denominator expression in the performance indicator definition corrects for double counting

[10] Starting point element $x_i^{[l]} = S_{i,l}$ for $i = 1, ..,$I and $l = 1, ..,$L

$$\begin{aligned}
\Pi_S^{[\mathrm{m}]} &= \frac{100\%}{\mathrm{L}} \sum_{l\in 1}^{\mathrm{L}} \frac{I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq 0, \gamma\in\{1,..,\underline{\mathrm{K}}\}, q_{\mathrm{E}}^{[\kappa]}=m\Big)}{\sum_{\gamma\in 1}^{\underline{\mathrm{K}}} I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) = f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big), q_{\mathrm{E}}^{[\kappa]}=m\Big)} \\
\Pi_S^{[\mathrm{Decom.}]} &= \frac{100\%}{\mathrm{L}} \sum_{l\in 1}^{\mathrm{L}} \frac{I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq 0, \gamma\in\{1,..,\underline{\mathrm{K}}\}, q_{\mathrm{D}}^{[\kappa]}=\mathrm{Yes}\Big)}{\sum_{\gamma\in 1}^{\underline{\mathrm{K}}} I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) = f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big), q_{\mathrm{D}}^{[\kappa]}=Yes\Big)} \\
\Pi_S^{[\mathrm{Comp.}]} &= \frac{100\%}{\mathrm{L}} \sum_{l\in 1}^{\mathrm{L}} \frac{I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq 0, \gamma\in\{1,..,\underline{\mathrm{K}}\}, q_{\mathrm{C}}^{[\kappa]}=\mathrm{Yes}\Big)}{\sum_{\gamma\in 1}^{\underline{\mathrm{K}}} I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) = f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big), q_{\mathrm{C}}^{[\kappa]}=Yes\Big)} \\
\Pi_S^{[\mathrm{M=400}]} &= \frac{100\%}{\mathrm{L}} \sum_{l\in 1}^{\mathrm{L}} \frac{I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq 0, \gamma\in\{1,..,\underline{\mathrm{K}}\}, q_{\mathrm{S}}^{[\kappa]}=\mathrm{M=400}\Big)}{\sum_{\gamma\in 1}^{\underline{\mathrm{K}}} I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) = f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big), q_{\mathrm{S}}^{[\kappa]}=\mathrm{M=400}\Big)} \\
\Pi_S^{[\mathrm{WS}]} &= \frac{100\%}{\mathrm{L}} \sum_{l\in 1}^{\mathrm{L}} \frac{I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big) \geq 0, \gamma\in\{1,..,\underline{\mathrm{K}}\}, q_{\mathrm{W}}^{[\kappa]}=\mathrm{Yes}\Big)}{\sum_{\gamma\in 1}^{\underline{\mathrm{K}}} I\Big(f_\kappa^{\mathrm{RPSolve}}\big(x^{[l]}\big) = f_\gamma^{\mathrm{RPSolve}}\big(x^{[l]}\big), q_{\mathrm{W}}^{[\kappa]}=Yes\Big)}
\end{aligned} \tag{4.37}$$

for estimation methods $m$ ∈{MC, SMC, N-1MC, DS, ES, Diamond, Ball, Cube}, given starting point elements $x^{[l]}$ from $S$. This means that for instance $\Pi_S^{[\mathrm{MC}]}$ is the percentage that the MC method is the best performing method from in total L experiments, irrespective of the parameter setting for Compression, Decomposition, Warm Start or the number of samples.

**Example 4.9** *We continue with the results of Example 4.8, with given starting points* $S = (x^{[1]}, x^{[2]}, x^{[3]})$*. It follows that* $\pi_1^{[11]} = 33\%$*,* $\pi_1^{[15]} = 33\%$*,* $\pi_1^{[59]} = 0\%$ *and* $\pi_1^{[63]} = 0\%$ *and for instance* $\Pi_1^{[SMC]} = 66\%$*,* $\Pi_1^{[DS]} = 0\%$ *and* $\Pi_1^{[Decom.]} = 33\%$*. The other settings, i.e. the compression method, the number of samples M and the Warm Start method are not varied in this example.*

### 4.6.1 Statistical inference about RP method performance indicators

In the previous section the performance indicators $\pi_S^{[\kappa]}$ and $\Pi_S^{[type]}$ are defined, respectively for RP method combination $\kappa = 1, .., \underline{\mathrm{K}}$ and RP method aggregation *type* ∈ $\mathbb{M}$={MC, SMC, N-1MC, DS, ES, Diamond, Cube, Ball, Decom., Comp., M, WS}, given a sample $S$ of random starting points. It is possible that the results as in Example 4.9, can be different for different matrices of starting points. In other words, the performance indicators are random variables[11] $\pi_{\boldsymbol{S}}^{[\kappa]}$ and $\Pi_{\boldsymbol{S}}^{[type]}$, given random starting points $\boldsymbol{S}$.

[11] For the situation that each starting point $x^{[l]}$ always leads to one best method $\kappa$, i.e. $f_\kappa^{\mathrm{RPSolve}}\left(x^{[l]}\right) > f_\gamma^{\mathrm{RPSolve}}\left(x^{[l]}\right)$, $\forall\gamma \neq \kappa$, then $\frac{\mathrm{L}}{100}(\pi_{\boldsymbol{S}}^{[1]}, .., \pi_{\boldsymbol{S}}^{[\kappa]}, .., \pi_{\boldsymbol{S}}^{[K]})$ follows a multinomial distribution

The goal of RP method performance testing, is to conclude which RP methods perform above average. Statistical inference is done to make such conclusions in a robust way. Let $\mu_{\pi^{[\kappa]}}$ and $\mu_{\Pi^{[type]}}$ be the expected values of $\pi^{[\kappa]}_{\boldsymbol{S}}$ and $\Pi^{[type]}_{\boldsymbol{S}}$ respectively.

For statistical inference about the expected value $\mu_{\pi^{[\kappa]}}$ of the RP method performance indicator we have the following. Let us assume that all $\underline{\mathrm{K}}$ methods perform equally well, then one would expect $\mu_{\pi^{[\kappa]}} = \frac{100\%}{\underline{\mathrm{K}}}$ for $\kappa = 1, .., \underline{\mathrm{K}}$. Hence, a method is performing above average if the performance indicator is significantly above $\frac{100\%}{\underline{\mathrm{K}}}$. Therefore we focus on the hypothesis

$$\begin{aligned} H_0 :& \mu_{\pi^{[\kappa]}} = \frac{100\%}{\underline{\mathrm{K}}} \\ H_a :& \mu_{\pi^{[\kappa]}} > \frac{100\%}{\underline{\mathrm{K}}} \end{aligned} \tag{4.38}$$

Showing that $H_0$ does not hold, implies that the performance of method $q^{[\kappa]}$ is above average beyond reasonable doubt[12].

Consider the expected value $\mu_{\Pi^{[type]}}$ of the aggregated RP method performance indicator. If $\boldsymbol{v}$ is Normally distributed and the Happy set is polyhedral, then only 6 estimation methods $m$ with $m \in \{\mathrm{MC, SMC, N\text{-}1MC, DS, Ball, Cube}\}$ are feasible. If all methods perform equally well, then one would expect $\mu_{\Pi^{[m]}} = \frac{100\%}{6}$. Therefore we focus on the hypothesis

$$\begin{aligned} H_0 :& \mu_{\Pi^{[m]}} = \frac{100\%}{6} \\ H_a :& \mu_{\Pi^{[m]}} > \frac{100\%}{6} \end{aligned} \tag{4.39}$$

Showing that $H_0$ does not hold for method $m$, implies that the performance of this method is above average beyond reasonable doubt.

The other RP methods in $\{\mathrm{Comp., Decom., M} = 400, \mathrm{WS}\}$ are all either switched on or off. This means that if for instance Compression does not influence the performance, then one would expect $\mu_{\Pi^{[Comp.]}} = 50\%$. On the other if $\mu_{\Pi^{[Comp.]}} > 50\%$ then the Compression method has a positive influence on Robustness optimisation and a negative influence if $\mu_{\Pi^{[Comp.]}} < 50\%$. Therefore it is relevant to verify the hypothesis

$$\begin{aligned} H_0 :& \mu_{\Pi^{[type]}} = 50\% \\ H_a :& \mu_{\Pi^{[type]}} > 50\% \end{aligned} \tag{4.40}$$

as well as verifying

$$\begin{aligned} H_0 :& \mu_{\Pi^{[type]}} = 50\% \\ H_a :& \mu_{\Pi^{[type]}} < 50\% \end{aligned} \tag{4.41}$$

with success probabilities $\frac{1}{100}(\mu_{\pi^{[1]}}, .., \mu_{\pi^{[\kappa]}}, .., \mu_{\pi^{[K]}})$. However, this probability distribution does not apply, when more than one best method for a given starting point exists.

[12] The concepts of hypothesis testing are for instance discussed by Ott and Longnecker (2001). The significance level is by convention set to 0.05 and means that if the likelihood of $H_0$ being true is less than 5%, then $H_0$ is rejected and we say that $H_a$ is true beyond reasonable doubt.

for performance indicator (4.43).

Statistical inference about the expected values of the performance indictors is based on computing averages in the following way. Let $\boldsymbol{S}^{[1]}, .., \boldsymbol{S}^{[\mathrm{D}]}$ be i.i.d. as $\boldsymbol{S}$. Consequently, the random average performance indicators are

$$\boldsymbol{\pi}^{[\kappa]} = \frac{1}{\mathrm{D}} \sum_{d=1}^{\mathrm{D}} \pi^{[\kappa]}_{\boldsymbol{S}^{[d]}} \tag{4.42}$$

$$\boldsymbol{\Pi}^{[type]} = \frac{1}{\mathrm{D}} \sum_{d=1}^{\mathrm{D}} \Pi^{[type]}_{\boldsymbol{S}^{[d]}} \tag{4.43}$$

with corresponding realisations $\pi^{[\kappa]}$ and $\Pi^{[type]}$ for given starting point samples $S^{[1]}, .., S^{[\mathrm{D}]}$.

From the Central Limit Theorem follows that $\boldsymbol{\pi}^{[\kappa]}$ and $\boldsymbol{\pi}^{[type]}$ are Normally distributed for D → ∞. According Ott and Longnecker (2001) it is common to assume a student $t$-distributed test statistic, when hypothesis testing a mean value based on an estimated variance and leads to the following approach. The test statistics are $\boldsymbol{\tau}^{[\kappa,\mu]} = \frac{\boldsymbol{\pi}^{[\kappa]}-\mu}{\boldsymbol{s}_{[\kappa]}} \sim \boldsymbol{t}(\mathrm{D}-1)$ and $\boldsymbol{\tau}^{[type,\mu]} = \frac{\boldsymbol{\Pi}^{[type]}-\mu}{\boldsymbol{s}_{[type]}} \sim \boldsymbol{t}(\mathrm{D}-1)$, where $\boldsymbol{t}(\mathrm{D}-1)$ is a t-distributed random variable with D-1 degrees of freedom, where the variances are estimated with:

$$s^2_{[\kappa]} = v\hat{a}r(\boldsymbol{\pi}^{[\kappa]}) = \frac{1}{\mathrm{D}} \frac{1}{\mathrm{D}-1} \sum_{d=1}^{\mathrm{D}} \left( \pi^{[\kappa]}_{S^{[d]}} - \pi^{[\kappa]} \right)^2 \tag{4.44}$$

$$s^2_{[type]} = v\hat{a}r(\boldsymbol{\Pi}^{[type]}) = \frac{1}{\mathrm{D}} \frac{1}{\mathrm{D}-1} \sum_{d=1}^{\mathrm{D}} \left( \Pi^{[type]}_{S^{[d]}} - \Pi^{[type]} \right)^2 \tag{4.45}$$

given the observations $\pi^{[\kappa]}_{S^{[d]}}$ and $\Pi^{[type]}_{S^{[d]}}$ for $d = 1, ..,\mathrm{D}$. The p-values corresponding the hypothesis (4.38), (4.39), (4.40) and (4.41) are respectively:

$$\Pr\left\{ \boldsymbol{t}(\mathrm{D}-1) > \tau^{\left[\kappa, \frac{100\%}{\mathrm{K}}\right]} \right\} \tag{4.46}$$

$$\Pr\left\{ \boldsymbol{t}(\mathrm{D}-1) > \tau^{\left[type, \frac{100\%}{6}\right]} \right\} \tag{4.47}$$

$$\Pr\left\{ \boldsymbol{t}(\mathrm{D}-1) > \tau^{[type,50\%]} \right\} \tag{4.48}$$

$$\Pr\left\{ \boldsymbol{t}(\mathrm{D}-1) < \tau^{[type,50\%]} \right\} \tag{4.49}$$

where $\tau^{\left[\kappa, \frac{100\%}{\mathrm{K}}\right]}$, $\tau^{\left[type, \frac{100\%}{6}\right]}$ and $\tau^{[type,50\%]}$ are the observed test statistics.

**Algorithm 4.1** $\hat{R} = f_{\kappa}^{\mathrm{RPEst}}(x)$, where $\mathrm{q}^{[\kappa]}$=($\mathrm{q_E}$,$\mathrm{q_C}$,$\mathrm{q_D}$,$\mathrm{q_W}$,$\mathrm{q_S}$) with Table 4.1 values

```
PARAMETER: The Happy set $\mathbb{H}(x)$
PARAMETER: Probability distribution type of each element $\boldsymbol{v}_n$
PARAMETER: Matrix $V$, of M=400 samples of $\boldsymbol{v}$
Redefine the Happy set according Corollary 4.1 (on page 86) if necessary
if decomposition method is switched on (i.e $\mathrm{q_D}$=Yes) then
    Decompose $\mathbb{H}(x)$ into $\mathbb{H}_g(x)$ for $g = 1,..,$G. Also gives the index set $\mathbb{I}_g(x)$, that tells us which elements of $\boldsymbol{v}$ are associated to $\mathbb{H}_g(x)$
else
    G:=1
end if
Determine bounding approach. if $\mathrm{q_E}$=Diamond then $p := 1$ elseif $\mathrm{q_E}$=Ball then $p := 2$ elseif $\mathrm{q_E}$=Cube then $p := \infty$ end if
for $g = 1$ to G do
    Compute the radius $\underline{r}_g^{[p]}(x)$ of the p-norm based bounding set in $\mathbb{H}_g(x)$. If $p \neq 2$ and the Warm Start method is on, then also compute $\underline{r}_g^{[2]}(x)$, .
    if Compression method is switched on (i.e. $\mathrm{q_C}$=Yes) then
        Try compress $\mathbb{H}_g(x)$, given the probability distribution of $\boldsymbol{v}_n$ with $n \in \mathbb{I}_g(x)$.
    end if
end for
if $\mathrm{q_E} \in \{$Diamond, Ball, Cube$\}$ then
    Determine lower bound $\hat{R} = \prod_{g=1}^{\mathrm{G}} \Pr\left\{\|\boldsymbol{v}\|_p \leq \underline{r}_g^{[p]}(x)\right\}$
    if $\hat{R} = 0$ and Warm Start is on ($\mathrm{q_W}$=Yes) then $\hat{R} := \min_g \underline{r}_g^{[2]}(x)$ end
    EXIT
else
    for all samples ($m$ =1 to $\mathrm{q_S}$) and all decomposed groups (g=1 to G) do
        compute $\hat{r}_{m,g}$, which is an estimate of $\Pr\left\{\boldsymbol{v}^{[g]} \in \mathbb{H}_g(x)\right\}$ based on estimation method E and sample (column) $m$ in V
    end for
end if
Determine the Robustness estimate $\hat{R} = \prod_{g=1}^{\mathrm{G}} \frac{1}{\mathrm{q_S}} \sum_{m=1}^{\mathrm{q_S}} \hat{r}_{m,g}$
if $\hat{R} = 0$ and Warm Start is on then
    $\hat{R} = \min_g \underline{r}_g^{[2]}(x)$
end if
```

**Algorithm 4.2** $\tilde{R}\left(x^{[l,\kappa]*}\right) = f_\kappa^{\mathrm{RPSolve}}(x^{[l]})$

```
PARAMETER: $\mathbb{X}$, the set of deterministic feasible designs;
Start solving $\max_{x \in \mathbb{X}} \left[f_\kappa^{\mathrm{RPEst}}(x)\right]$ with starting point $x^{[l]}$.
while solver is running do
    Determine the number of iterations ($i$) finished
    Determine the best feasible design $x^*$ corresponding the highest Robustness estimate $f_q^{\mathrm{RPEst}}(x^*)$
    found until iteration $i$.
    if $i \leq 20$ then $x^{[l,\kappa]*} := x^*$
    if $i = 20$ then stop solver
end while
Compute accurate estimate $\tilde{R}\left(x^{[l,\kappa]*}\right)$ based on 20000 samples, given best design $x^{[l,\kappa]*}$ (See Appendix
A.3).
```

## 4.7 Concluding remarks

The RP methods discussed in Chapter 3 are Robustness estimation methods and are either based on sampling (MC, SMC, N-1MC, DS and ES) or on bounding (Diamond, Ball and Cube). The RP methods discussed in Chapter 4 are Robustness computation methods for improving the efficiency and effectiveness of Robustness estimation methods. The Robustness computation methods are respectively called the Compression method, the Decomposition method and the Warm Start method. Furthermore, a methodology is given in Chapter 4 for comparing the performance of RP methods in the context of Robustness optimisation. The RP methods of Chapter 3 and Chapter 4 are applied in case studies, which are discussed in Chapter 5.

# Chapter 5
# Case Studies

## 5.1 Introduction

In this chapter, the performance of the Robustness Programming methods as given in Chapters 3 and 4, are measured by numerical experiments based on three case studies. The first case is described in Section 5.2 and deals with optimal robust mixture design of a food product. The second case is given in 5.4 and deals with optimal robust raw material cost planning in a food production context. The last case is outlined in Section 5.5 and deals with coalition robustness in a two-stage cartel game, in the context of $CO_2$ reduction strategies in climate agreements. Each of these sections first introduces the case and gives a formulation for the corresponding Robustness Programming problem, followed by an elaboration on the performance statistics of the RP methods.

## 5.2 Optimal Robust Mixture Design

Together with Unilever R&D, research has been carried out on Mixture Design Problems for food products. The *original* mixture design case is based on confidential data. To sketch the robust Mixture Design Problem and the application of the methods for Robustness Programming, a *hypothetical* case is used, which is identical to the original case in terms of stochastic models, dimensions, polynomial degree and number of constraints.

In Section 5.2.1 the model for defining a robust mixture design for a single product is introduced. In Section 5.2.2 this model is extended for defining a robust mixture design for multiple products simultaneously. Finally, the results of comparing the RP methods for solving the original multiple product mixture design case, are given in Section 5.2.3.

### 5.2.1 Single Product Robust Mixture Design

The challenge is to find Raw Material (RM) proportions $(x_1, .., x_\mathrm{I})^\intercal$, such that product specifications for protein $(p)$, carbohydrate $(c)$, fat $(f)$ and bacteria growth rates at 0°C, 4°C, 10°C and 22°C $(b_0, b_4, b_{10}, b_{22})$, are satisfied. The vector $x$ is called the *product design.* In the Unilever case there are 11 RM's to choose from, i.e. $\mathrm{I} = 11$. The specifications are defined as intervals:

$$\begin{gathered}
[{}^{p}Low, {}^{p}High] \\
[{}^{c}Low, {}^{c}High] \\
[{}^{f}Low, {}^{f}High] \\
[{}^{b_0}Low, {}^{b_0}High] \\
[{}^{b_4}Low, {}^{b_4}High] \\
[{}^{b_{10}}Low, {}^{b_{10}}High] \\
[{}^{b_{22}}Low, {}^{b_{22}}High]
\end{gathered} \tag{5.1}$$

An analytical model for predicting the nutritional properties as a function of raw material proportions is defined as:

$$\begin{bmatrix} {}^{p}y \\ {}^{c}y \\ {}^{f}y \end{bmatrix} = Ax \tag{5.2}$$

where ${}^{p}y$,${}^{c}y$ and ${}^{f}y$ are respectively the weight fractions of protein, carbohydrates and fat in the product. Food scientists consider this linear model as *acceptably* accurate and concluded that there is no reason to study the Robustness of the nutritional model. In connection to the Robustness Programming Framework, the nutritional model is typically part of the deterministic object properties. The set of *deterministic feasible designs* for this case is defined as:

$$\mathbb{X} = \left\{ x \in \mathbb{R}^{\mathrm{I}} \middle| \begin{bmatrix} {}^{p}Low \\ {}^{c}Low \\ {}^{f}Low \end{bmatrix} \leq Ax \leq \begin{bmatrix} {}^{p}High \\ {}^{c}High \\ {}^{f}High \end{bmatrix} ; \sum_{i=1}^{\mathrm{I}} x_i = 1; x_i \geq 0 \ , \ i = 1, .., \mathrm{I} \right\} \tag{5.3}$$

Food scientists developed regression models for predicting the bacteria growth rates as a function of specific components in the raw materials. These bacteria growth rate models are based on the results of experiments, which inevitably have some degree of uncertainty. Consequently, one cannot be absolutely sure about the model prediction and in particular whether or not the specifications are satisfied. In this context it is relevant to find such raw material proportions, that the probability of satisfying the bacteria growth rate specifications, is maximised. In the following, a regression model for predicting the bacteria growth rates is introduced in relation to the specification Robustness.

It has been decided to base the regression model on the proportions of 16 categories of nutritional components in the raw materials instead of the raw material proportions. The 16 categories of nutritional components in the raw materials are called *Growth-rate Affecting Component*s (GACs). Examples of GACs are various categories of amino acids, fats and carbohydrates. This model assumes that the influence of each raw material on the bacteria growth rate, can be characterised by the proportions of these GACs. This way it is possible to study the effect of a new raw material, when only given the GAC specification from the supplier.

The model for predicting bacteria growth-rates has two stages: In the first stage, the weight-fraction of each GAC in the mixture of raw materials is determined, given the product design $x$. In the second stage, a regression model predicts the bacteria growth-rates at the four different temperatures, given the GAC weight fractions.

The relation between raw material weight proportions ($x$) and the weight proportions of GACs ($z$), is defined by an L×I matrix $G$ with $z = Gx$, where L= 16 and I=11. Food scientists concluded that the value of $G$ is known with sufficient accuracy in practice. No uncertainty about $G$ is considered in this model. For example if only RM 1 and RM 11 are used, with relative weight proportions $x_1 = \frac{1}{3}$ and $x_{11} = \frac{2}{3}$, then $z_l = \frac{1}{3}G_{l,1} + \frac{2}{3}G_{l,11}$ for $l = 1,..,$L and is illustrated in Figure 5.1, where each arc from $x_i$ to $z_l$ symbolises the fraction $G_{l,i}$ of GAC $l$ in raw material $i$.

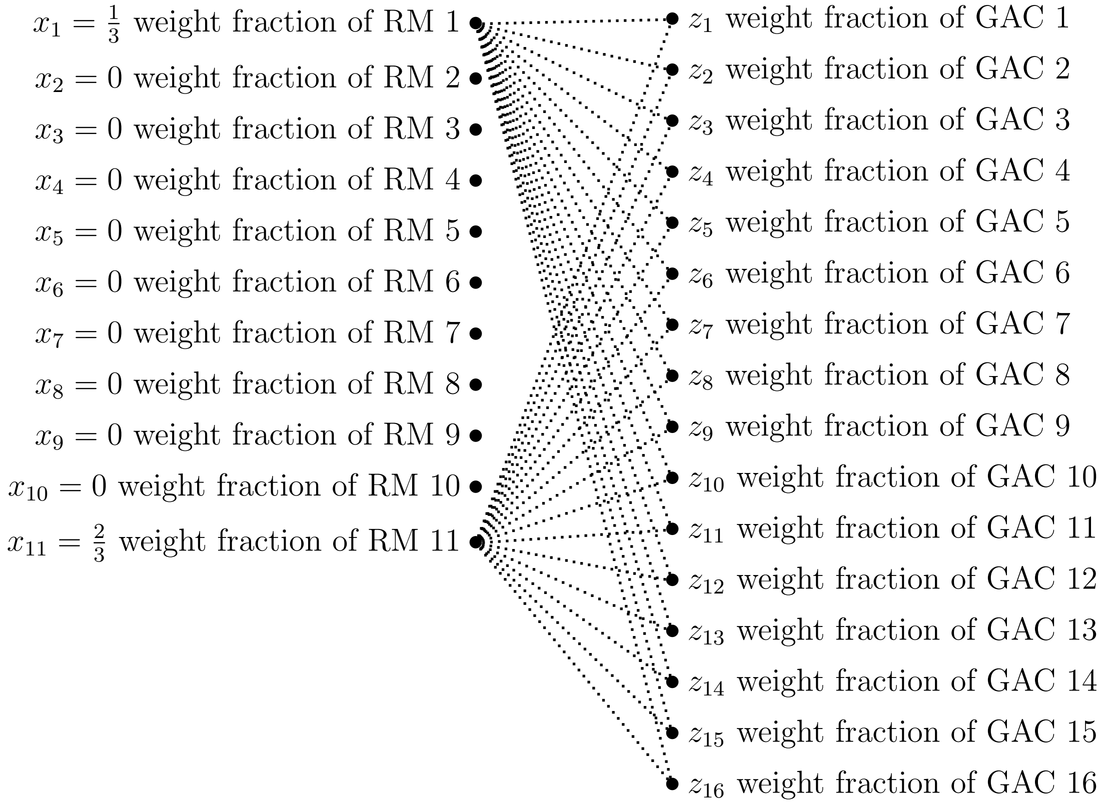

$x_1 = \frac{1}{3}$ weight fraction of RM 1
$x_2 = 0$ weight fraction of RM 2
$x_3 = 0$ weight fraction of RM 3
$x_4 = 0$ weight fraction of RM 4
$x_5 = 0$ weight fraction of RM 5
$x_6 = 0$ weight fraction of RM 6
$x_7 = 0$ weight fraction of RM 7
$x_8 = 0$ weight fraction of RM 8
$x_9 = 0$ weight fraction of RM 9
$x_{10} = 0$ weight fraction of RM 10
$x_{11} = \frac{2}{3}$ weight fraction of RM 11
$z_1$ weight fraction of GAC 1
$z_2$ weight fraction of GAC 2
$z_3$ weight fraction of GAC 3
$z_4$ weight fraction of GAC 4
$z_5$ weight fraction of GAC 5
$z_6$ weight fraction of GAC 6
$z_7$ weight fraction of GAC 7
$z_8$ weight fraction of GAC 8
$z_9$ weight fraction of GAC 9
$z_{10}$ weight fraction of GAC 10
$z_{11}$ weight fraction of GAC 11
$z_{12}$ weight fraction of GAC 12
$z_{13}$ weight fraction of GAC 13
$z_{14}$ weight fraction of GAC 14
$z_{15}$ weight fraction of GAC 15
$z_{16}$ weight fraction of GAC 16

Figure 5.1: Illustration of $Gx = z$

Food-scientists have determined that both the individual contribution of each GAC, as well as the combination of GAC's influence bacteria growth-rates. Therefore, the regression model is based on a function $f(Gx)$ which models both the linear effect and interaction effects of GACs on bacteria growth-rates. The function $f$ will be explained hereafter. The regression model has the following form:

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \boldsymbol{y}_4 \end{bmatrix} = f(Gx)^{\intercal}\beta + \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \boldsymbol{e}_3 \\ \boldsymbol{e}_4 \end{bmatrix} \tag{5.4}$$

where $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, $\boldsymbol{y}_3$ and $\boldsymbol{y}_4$ respectively refer to the bacteria growth-rates at the 4 temperatures 0°C, 4°C, 10°C and 22°C. The linear effects and interaction effects are modelled as a matrix function: $f : \mathbb{R}^{\mathrm{L}} \rightarrow \mathbb{R}^{4 \times \mathrm{Z}}$, which returns a matrix with the following number of columns $\mathrm{Z} = 4\left(1 + \mathrm{L} + \frac{1}{2}(\mathrm{L}^2 - \mathrm{L})\right)$. I.e. this matrix refers to 1 intercept, L linear effects and $\frac{1}{2}(\mathrm{L}^2 - \mathrm{L})$ interaction effects for all four temperature specific bacteria growth rate models. As an illustration consider the case L = 2, then the matrix function is defined as:

$$f(Gx)^{\intercal} = \begin{bmatrix} 1 & z_1 & z_2 & z_1z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & z_1 & z_2 & z_1z_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & z_1 & z_2 & z_1z_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & z_1 & z_2 & z_1z_2 \end{bmatrix}$$

In the Unilever case, each raw material is considered to consist of L = 16 GAC components and consequently the size of vector $\beta$ is Z = 548 elements. The model assumes additive noise, where the errors $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \boldsymbol{e}_4$ are considered to be normally distributed random variables:

$$\boldsymbol{e} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \sigma_{1,3} & \sigma_{1,4} \\ \sigma_{2,1} & \sigma_2^2 & \sigma_{2,3} & \sigma_{2,4} \\ \sigma_{3,1} & \sigma_{3,2} & \sigma_3^2 & \sigma_{3,4} \\ \sigma_{4,1} & \sigma_{4,2} & \sigma_{4,3} & \sigma_4^2 \end{bmatrix}\right) = N(\mu^{(e)}, \Sigma^{(e)}) \tag{5.5}$$

In the original mixture design case, the errors were assumed to be independent and the estimation of $\beta$ was done with the OLS[1] regression technique. The consequence of independent errors is that $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ for $i \neq j$ are also independent. More importantly, it can be shown that such independency structure makes it possible to apply the decomposition method as introduced in Section 4.3.1. Therefore, to make results comparable, the vector $\beta$ is also estimated with OLS regression technique and the errors in this hypothetical case are assumed to be independent, which in the case of the Normal distribution translates to $\sigma_{i,j} = 0$ for $i \neq j$. Verbeek (2004) introduces a method called *Feasible Generalised Least Squares* (FGLS) which can be used to estimate $\beta$ based on an estimated $\Sigma^{(e)}$ in the situation that the elements of the error vector are dependent.

The bacteria growth rate model is estimated in the following way. The experimental design matrix $F$ for M = 1058 different combinations of raw materials, is defined as:

$$F = \begin{bmatrix} f(z^{[1]})^\intercal \\ \vdots \\ f(z^{[m]})^\intercal \\ \vdots \\ f(z^{[\mathrm{M}]})^\intercal \end{bmatrix} \tag{5.6}$$

where $z^{[\mathrm{m}]} = Gx^{[\mathrm{m}]}$ is the $m$-th combination of GAC's, given the raw material proportion $x^{[\mathrm{m}]}$ for $m = 1, .., \mathrm{M}$. Matrix $F$ has 4M rows and 4Z columns. The bacteria growth-rate measurements are collected in the vector

$$Y = \begin{bmatrix} y^{[1]} \\ \vdots \\ y^{[m]} \\ \vdots \\ y^{[\mathrm{M}]} \end{bmatrix}$$

where $y^{[\mathrm{m}]}$ is a vector of 4 elements, referring to the 4 temperature measurements and vector $Y$ has 4M elements. Similarly, the 4 error elements in vector $e^{[m]}$ connected to the $m$-th RM combination can be *stacked* in a vector $E = (e_1^{[1]}, e_2^{[1]}, e_3^{[1]}, e_4^{[1]}, e_1^{[2]}, ..., e_4^{[\mathrm{M}]})^\intercal$ with 4M elements. Consequently,

$$\boldsymbol{E} \sim N\left([0], \Sigma^{[E]}\right)$$

with [0] a vector with 4M elements which all have the value 0 and

$$\Sigma^{[E]}_{4m+i,4m+i} = \Sigma^{[e]}_{i,i}$$

[1] Ordinary Least Squares is a method for estimating regression parameters (see e.g. Rice, 1995).

for $i = 1, .., 4$, $m = 0, .., \mathrm{M}-1$ and where all non-diagonal elements are zero. The regression model, given the M experiments defined by $F$, is:

$$\boldsymbol{Y} = F\beta + \boldsymbol{E}$$

Rice (1995) explains that the OLS estimator of $\beta$ is

$$\hat{\boldsymbol{\beta}} = (F'F)^{-1}F'\boldsymbol{Y}$$

and follows the normal distribution, which can be expressed as

$$\hat{\boldsymbol{\beta}} \sim N\left(\beta, \left(F^{\intercal}(\Sigma^{[E]})^{-1}F\right)^{-1}\right)$$

The estimate $\hat{\beta} = (F'F)^{-1}F'Y$ can be computed, given the measurements $Y$. The growth rate estimates are $\hat{Y} = F\hat{\beta}$ and can be used to estimate the error variances on the diagonal of $\Sigma^{[e]}$ with

$$\hat{\Sigma}^{[e]}_{i,i} = \hat{\sigma}^2_i = \frac{1}{\text{M-}\frac{1}{4}\text{Z}} \sum_{m=1}^{\text{M}} (Y_{i+4(m-1)} - \hat{Y}_{i+4(m-1)})^2$$

for $i = 1, .., 4$ and gives the variance estimate of the elements of $\boldsymbol{E}$ with

$$\hat{\Sigma}^{[E]}_{4m+i,4m+i} = \hat{\Sigma}^{[e]}_{i,i}$$

for $i = 1, .., 4$, $m = 0, .., \mathrm{M} - 1$ where all non-diagonal elements are zero.

The stochastic model of the estimator can be used to express the *confidence* that a model prediction is within the specification. Analogously to the definition in Ott and Longnecker (2001) about a *prediction interval* and *prediction level* for single-response regression models ($y \in \mathbb{R}$), the following *prediction level* ($\alpha$) can be defined for the multi-response ($y \in \mathbb{R}^4$) bacteria growth rate model where the prediction interval is the Happy set defined by the product specifications:

$$\alpha = R(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}(x)\right\} \tag{5.7}$$

with

$$\mathbb{H}(x) = \left\{ v \in \mathbb{R}^{\mathrm{Z}+4} \,\middle\|\, \begin{bmatrix} {}^{b_0}Low \\ {}^{b_4}Low \\ {}^{b_{10}}Low \\ {}^{b_{22}}Low \end{bmatrix} \leq \begin{bmatrix} u_1(x,v) \\ u_2(x,v) \\ u_3(x,v) \\ u_4(x,v) \end{bmatrix} \leq \begin{bmatrix} {}^{b_0}High \\ {}^{b_4}High \\ {}^{b_{10}}High \\ {}^{b_{22}}High \end{bmatrix} \right\} \tag{5.8}$$

and

$$\begin{bmatrix} u_1(x,v) \\ u_2(x,v) \\ u_3(x,v) \\ u_4(x,v) \end{bmatrix} = f(Gx)^{\intercal}\hat{\beta} + \begin{bmatrix} f(Gx) \\ I \end{bmatrix}^{\intercal} Tv \tag{5.9}$$

where the square matrix $T$ is obtained by Cholesky decomposition such that

$$TT^{\intercal} = \begin{bmatrix} \left(F^{\intercal}(\hat{\Sigma}^{[E]})^{-1}F\right)^{-1} & [0]^{\intercal} \\ [0] & \hat{\Sigma}^{[e]} \end{bmatrix} \tag{5.10}$$

and [0] is a 4×Z zero matrix and $I$ is a 4×4 identity matrix. The dimensions, matrices and vectors described in the model for the single product design case, are summarised in Table 5.1.

**Table 5.1**: Overview of symbols in single product design case

| name | type | size | description | details |
|---|---|---|---|---|
| I | dimension | 11 | Number of raw materials | associated index: $i$ |
| L | dimension | 16 | Number of GACs | associated index: $l$ |
| M | dimension | 1058 | Number of experiments per temperature | associated index: $m$ |
| Z | dimension | 548 | Number of reg. coefficients $\beta$ | |
| N | dimension | Z+4 | Number of RP random vector | |
| $T$ | constant matrix | (Z+4)×(Z+4) | Cholesky decomposition of covariance matrix | see (5.10) |
| $F$ | constant matrix | M×Z | Design of experiments | see (5.6) |
| $G$ | constant matrix | L×I | GAC weight fractions in RM | $z = Gx$ |
| $\boldsymbol{v}$ | random vector | N=Z+4 | t-distributed random vector | |
| $x$ | vector | I | product design vector | |
| $\hat{\beta}$ | constant vector | Z+4 | regression coefficients vector | |

It can be shown that $u_i(x, \boldsymbol{v})$ and $u_j(x, \boldsymbol{v})$ for $i \neq j$ are independent, because the error elements in $\boldsymbol{E}$ are independent. It can be shown that the random vector $\boldsymbol{v}$ follows a $\boldsymbol{t}(r)$ distribution[2] with $r$ =M-$\frac{1}{4}$Z degrees of freedom, because the variances of the errors are estimated.

The statistical *level of confidence* measure $\alpha$, quantifies the confidence that an individual product will be according to specifications. This model takes into account 2 sources of uncertainty

1. The uncertainty of natural variation in growth rates ($\boldsymbol{e}$).

2. The uncertainty in the mean value predictions of the model ($f(x)^{\intercal}\boldsymbol{\beta}$)

Methods for Robustness Programming can be used to find a product design $x^*$, with optimal confidence level $R^*$ by solving (2.8) for the above described case. Product design vector $x^*$ is feasible (i.e. $x^* \in \mathbb{X}$) and corresponds the Robustness optimum $R^* = R(x^*)$, where each element $x_i^*$, $i = 1,..,$I is the weight fraction of raw material $i$ in the product.

[2] NB: $\boldsymbol{t}_n(r) \sim \frac{\boldsymbol{v}_n}{\sqrt{\frac{1}{r}\boldsymbol{\chi}^2(r)}}$ where $\boldsymbol{v}_n \sim N(0,1)$ and independent for $n = 1,..,$N and independent of the chi-square random variable $\boldsymbol{\chi}^2(r)$ with $r$ degrees of freedom. This means that for all $\boldsymbol{t}_n(r)$, $n = 1,..,$N the denominator is identical.

### 5.2.2 Multiple Product Robust Mixture Design

In this section the single product robust mixture design model of Section 5.2.1 is extended to the robust design of K products simultaneously. In the multiple product situation, the product specifications are

$$
\begin{gathered}
[{}^{p}Low_k, {}^{p}High_k] \\
[{}^{c}Low_k, {}^{c}High_k] \\
[{}^{f}Low_k, {}^{f}High_k] \\
[{}^{b_0}Low_k, {}^{b_0}High_k] \\
[{}^{b_4}Low_k, {}^{b_4}High_k] \\
[{}^{b_{10}}Low_k, {}^{b_{10}}High_k] \\
[{}^{b_{22}}Low_k, {}^{b_{22}}High_k]
\end{gathered}
\tag{5.11}
$$

for $k = 1, .., \mathrm{K}$. Let $x^{[k]}$ be the product design of the $k$-th product. The design of K food products can be represented by a matrix

$$
X = \begin{bmatrix} \uparrow & & \uparrow & & \uparrow \\ x^{[1]} & \cdots & x^{[k]} & \cdots & x^{[\mathrm{K}]} \\ \downarrow & & \downarrow & & \downarrow \end{bmatrix}
$$

The set of *deterministic feasible designs* for a single product is given in (5.12). Similarly, this set for K food products is $\prod_{k=1}^{\mathrm{K}} \mathbb{X}_k$ with

$$
\mathbb{X}_k = \left\{ x \in \mathbb{R}^{\mathrm{I}} \;\middle|\; \begin{bmatrix} {}^{p}Low_k \\ {}^{c}Low_k \\ {}^{f}Low_k \end{bmatrix} \leq Ax \leq \begin{bmatrix} {}^{p}High_k \\ {}^{c}High_k \\ {}^{f}High_k \end{bmatrix} ; \sum_{i=1}^{\mathrm{I}} x_i = 1; x_i \geq 0 \;,\; i = 1, .., \mathrm{I} \right\}
\tag{5.12}
$$

The bacteria growth rate model is used for each product:

$$
\begin{bmatrix} u_{1+4(k-1)}(x,v) \\ u_{2+4(k-1)}(x,v) \\ u_{3+4(k-1)}(x,v) \\ u_{4+4(k-1)}(x,v) \end{bmatrix} = f(Gx)^{\intercal} \hat{\beta} + \begin{bmatrix} f(Gx) \\ I \end{bmatrix}^{\intercal} Tv
\tag{5.13}
$$

for $k = 1, ..,$K and leads to the Happy set definition in the multiple product design case

$$\mathbb{H}(X) = \left\{ v \in \mathbb{R}^{\mathrm{Z}+4} \;\middle|\; \begin{bmatrix} {}^{b_0}Low_1 \\ {}^{b_4}Low_1 \\ {}^{b_{10}}Low_1 \\ {}^{b_{22}}Low_1 \\ \vdots \\ {}^{b_0}Low_k \\ {}^{b_4}Low_k \\ {}^{b_{10}}Low_k \\ {}^{b_{22}}Low_k \\ \vdots \\ {}^{b_0}Low_{\mathrm{K}} \\ {}^{b_4}Low_{\mathrm{K}} \\ {}^{b_{10}}Low_{\mathrm{K}} \\ {}^{b_{22}}Low_{\mathrm{K}} \end{bmatrix} \leq \begin{bmatrix} u_1(x^{[1]}, v) \\ u_2(x^{[1]}, v) \\ u_3(x^{[1]}, v) \\ u_4(x^{[1]}, v) \\ \vdots \\ u_{1+4(k-1)}(x^{[k]}, v) \\ u_{2+4(k-1)}(x^{[k]}, v) \\ u_{3+4(k-1)}(x^{[k]}, v) \\ u_{4+4(k-1)}(x^{[k]}, v) \\ \vdots \\ u_{1+4(\mathrm{K}-1)}(x^{[\mathrm{K}]}, v) \\ u_{2+4(\mathrm{K}-1)}(x^{[\mathrm{K}]}, v) \\ u_{3+4(\mathrm{K}-1)}(x^{[\mathrm{K}]}, v) \\ u_{4+4(\mathrm{K}-1)}(x^{[\mathrm{K}]}, v) \end{bmatrix} \leq \begin{bmatrix} {}^{b_0}High_1 \\ {}^{b_4}High_1 \\ {}^{b_{10}}High_1 \\ {}^{b_{22}}High_1 \\ \vdots \\ {}^{b_0}High_k \\ {}^{b_4}High_k \\ {}^{b_{10}}High_k \\ {}^{b_{22}}High_k \\ \vdots \\ {}^{b_0}High_{\mathrm{K}} \\ {}^{b_4}High_{\mathrm{K}} \\ {}^{b_{10}}High_{\mathrm{K}} \\ {}^{b_{22}}High_{\mathrm{K}} \end{bmatrix} \right\} \tag{5.14}$$

In this model, for $k = 1, ..,$K, the elements $\hat{\beta}_1, .., \hat{\beta}_{137}$ are only relevant for $u_{1+4(k-1)}(x^{[k]}, v)$, the elements $\hat{\beta}_{138}, .., \hat{\beta}_{275}$ are only relevant for $u_{2+4(k-1)}(x^{[k]}, v)$, the elements $\hat{\beta}_{276}, .., \hat{\beta}_{412}$ are only relevant for $u_{3+4(k-1)}(x^{[k]}, v)$ and the elements $\hat{\beta}_{413}, .., \hat{\beta}_{548}$ are only relevant for $u_{4+4(k-1)}(x^{[k]}, v)$. The bacteria growth rate model $u$ has the following dependencies structure, because the elements of the error vector $\boldsymbol{E}$ are independent: Similar to the single product design case, we have that $u_{i+4(k-1)}(x^{[k]}, \boldsymbol{v})$ and $u_{j+4(k-1)}(x^{[k]}, \boldsymbol{v})$ are independent for $i \neq j$ and $k = 1, ..,$K. However, $u_{i+4(k-1)}(x^{[k]}, \boldsymbol{v})$ and $u_{i+4(l-1)}(x^{[l]}, \boldsymbol{v})$ are dependent[3] for all $i = 1, .., 4$ and $k, l = 1, ..,$K. It can be shown that, for any $x$ the Happy set $\mathbb{H}(x) \subseteq \mathbb{R}^{\mathrm{Z}+4}$ can be decomposed (at least) into i=1,..,4 Happy sets $\mathbb{H}_i(x) \subseteq \mathbb{R}^{\frac{1}{4}\mathrm{Z}+1}$.

### 5.2.3 Results

Methods for Robustness Programming are used to solve

$$R^* = \max_{X \in \prod_{k=1}^{\mathrm{K}} \mathbb{X}_k} \left[ \Pr\{\boldsymbol{v} \in \mathbb{H}(X)\} \right] \tag{5.15}$$

with the Matlab FMINCON solver. The performance indicators $\pi$ and $\Pi$, as introduced in (4.42) and (4.43) and the corresponding p-values (4.46), (4.47), (4.48) and (4.49) are determined for the multiple product mixture design case, for all applicable alternative RP methods. The non-zero $\pi$ and $\Pi$ results are respectively shown in Table 5.2 and Table 5.3, where the number of samples of $\boldsymbol{v}$ is M=100. The corresponding computation times are illustrated in 5.4.

These results show that the Warm Start method is effective, except for the SMC method. This led to a second experiment where the Warm Start method is used except

[3] The dependency for any $x$, follows from the observation that the elements $v_1$, $v_{138}$, $v_{275}$ and $v_{412}$ represent the intercepts of the regression model, which means that $u_{i+4(k-1)}(x^{[k]}, \boldsymbol{v})$ and $\boldsymbol{v}_{(i-1)137+1}$ are dependent for all $x^{[k]} \in \mathbb{R}^{\mathrm{I}}$, $k = 1, ..,$K and $i = 1, .., 4$.

for the SMC method and the number samples is either M=100 or M=400. The non-zero results corresponding to the performance indicators $\pi$ and $\Pi$, are respectively shown in Table 5.5 and in Table 5.6, for the case that the WS=Yes. In Section 5.2.4 the results are discussed and conclusions are formulated.

**Table 5.2**: Performance indicator of combined RP methods for problem (5.15)

| RP Method $q^{[\kappa]}$ | | | | $\pi^{[\kappa]}$ | p-value<br>$H_a : \mu_{\pi^{[\kappa]}} > \frac{100\%}{48}$ |
|---|---|---|---|---|---|
| Estimator | Comp. | Decom. | WS | | |
| DS | Yes | Yes | Yes | 18% | **0.0000** |
| N-1MC | Yes | Yes | Yes | 14% | **0.0001** |
| Cube | Yes | Yes | Yes | 11% | **0.0042** |
| N-1MC | No | Yes | Yes | 9% | **0.0045** |
| Ball | Yes | Yes | Yes | 8% | **0.0171** |
| SMC | Yes | Yes | No | 6% | 0.0626 |
| SMC | No | No | No | 5% | **0.0243** |
| Ball | Yes | No | Yes | 5% | 0.0634 |
| SMC | No | Yes | No | 5% | 0.0689 |
| DS | No | No | Yes | 4% | 0.1190 |
| DS | No | Yes | Yes | 4% | 0.1651 |
| SMC | No | No | Yes | 3% | 0.1164 |
| Cube | No | No | Yes | 3% | 0.2665 |
| SMC | Yes | Yes | Yes | 3% | 0.1507 |
| SMC | No | Yes | Yes | 1% | 0.8796 |

Note a) The results for $\pi > 0$ are given in descending order of $\pi$.

Note b) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note c) M=100 number of samples of $\boldsymbol{v}$ and D=30 strata of L=4 randomly chosen starting points.

**Table 5.3**: Performance indicator of individual RP methods for problem (5.15)

| RP Method *type* | $\Pi^{[type]}$ | p-value<br>$H_a : \mu_{\Pi^{[type]}} > \frac{100\%}{6}$ | p-value<br>$H_a : \mu_{\Pi^{[type]}} > 50\%$ | p-value<br>$H_a : \mu_{\Pi^{[type]}} < 50\%$ |
|---|---|---|---|---|
| MC | 0.0% | 1.0000 | | |
| SMC | 23.3% | **0.0439** | | |
| N-1MC | 23.3% | 0.0515 | | |
| DS | 26.7% | **0.0086** | | |
| Ball | 12.5% | 0.9212 | | |
| Cube | 14.2% | 0.7268 | | |
| Compression (no MC/SMC) | 69.2% | | **0.0000** | 1.0000 |
| Decomposition (no MC/SMC) | 75.8% | | **0.0000** | 1.0000 |
| Warmstart | 84.6% | | **0.0000** | 1.0000 |
| Warmstart (no MC/SMC) | 100.0% | | **0.0000** | 1 |

Note a) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note b) M=100 number of samples of $\boldsymbol{v}$ and D=30 strata of L=4 randomly chosen starting points.

**Table 5.4**: Average computation time (sec.) for problem (5.15)

| | Compression | | Decomposition | | Warmstart | | |
|---|---|---|---|---|---|---|---|
| | No | Yes | No | Yes | No | Yes | mean |
| MC | | $132^{d}$ | $132^{d}$ | | | $132^{d}$ | 132 |
| SMC | 134 | 142 | 127 | 144 | 138 | 137 | 137 |
| N-1MC | 180 | 155 | 135 | 175 | | 165 | 162 |
| DS | 163 | 148 | 143 | 167 | | 155 | 155 |
| Ball | 134 | 125 | 122 | 131 | | 125 | 127 |
| Cube | 124 | 126 | 123 | 129 | | 126 | 126 |
| mean | 147 | 138 | 130 | 149 | 138 | 140 | |

Note a) Experiment is based on running all RP methods, given 10 starting points, with M=100 sample size

Note b) Computation is done on a 2GHz Intel Core Duo computer with 2GB memory

Note c) Time measurements are only given if all 20 iterations are completed and void spaces otherwise.

Note d) For these results the WS method did not result in finding a positive Robustness estimate within 20 iterations

**Table 5.5**: Performance indicator of combined RP methods for problem (5.15)

| RP Method $q^{[\kappa]}$ | | | | $\pi^{[\kappa]}$ | p-value |
|---|---|---|---|---|---|
| | | | | | $H_a : \mu_{\pi^{[\kappa]}} > \frac{100\%}{40}$ |
| Estimator | Comp. | Decom. | sample size | | |
| DS | Yes | Yes | 400 | 15% | **0.0007** |
| DS | Yes | Yes | 100 | 9% | **0.0093** |
| Cube | Yes | Yes | N.A. | 9% | **0.0187** |
| N-1MC | Yes | Yes | 400 | 8% | **0.0183** |
| SMC | No | Yes | 100 | 6% | **0.0665** |
| DS | No | Yes | 400 | 6% | 0.1016 |
| SMC | No | No | 100 | 5% | 0.1226 |
| N-1MC | No | Yes | 400 | 5% | 0.1682 |
| SMC | Yes | Yes | 100 | 5% | 0.1226 |
| SMC | Yes | Yes | 400 | 5% | 0.1226 |
| Ball | Yes | Yes | N.A. | 5% | 0.1682 |
| Cube | No | No | N.A. | 4% | 0.2689 |
| N-1MC | Yes | Yes | 100 | 4% | 0.2689 |
| DS | No | No | 100 | 3% | 0.3849 |
| SMC | No | No | 400 | 3% | 0.3849 |
| Ball | Yes | No | N.A. | 3% | 0.3849 |
| N-1MC | No | Yes | 100 | 2% | 0.6378 |
| DS | No | Yes | 100 | 1% | 0.9265 |
| SMC | No | Yes | 400 | 1% | 0.9265 |
| Cube | Yes | No | N.A. | 1% | 0.9265 |

Note a) The results for $\pi > 0$ are given in descending order of $\pi$.

Note b) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note c) WS=Yes (except for SMC) and D=33 strata of L=3 randomly chosen starting points.

**Table 5.6**: Performance indicator of individual RP methods for problem (5.15)

| RP Method *type* | $\Pi^{[type]}$ | p-value<br>$H_a : \mu_{\Pi^{[type]}} > \frac{100\%}{6}$ | p-value<br>$H_a : \mu_{\Pi^{[type]}} > 50\%$ | p-value<br>$H_a : \mu_{\Pi^{[type]}} < 50\%$ |
|---|---|---|---|---|
| MC | 0.0% | 1.0000 | | |
| SMC | 25.0% | **0.0465** | | |
| N-1MC | 19.0% | 0.2741 | | |
| DS | 34.0% | **0.0015** | | |
| Ball | 8.0% | 0.9973 | | |
| Cube | 14.0% | 0.7703 | | |
| Compression (no MC/SMC) | 64.0% | | **0.0063** | 0.9937 |
| Decomposition (no MC/SMC) | 76.0% | | **0.0000** | 1.0000 |
| M=400 (no bounding) | 51.0% | | 0.4362 | 0.5638 |

Note a) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note b) WS=Yes (except for SMC) and D=33 strata of L=3 randomly chosen starting points.

**Table 5.7**: Average computation time (sec.) for problem (5.15)

| | Compression | | Decomposition | | Sample size | | |
|---|---|---|---|---|---|---|---|
| | No | Yes | No | Yes | 100 | 400 | mean |
| MC | | 128$^d$ | 128$^d$ | | 130$^d$ | 127$^d$ | 128 |
| SMC | 151 | 147 | 133 | 162 | 139 | 171 | 150 |
| N-1MC | 237 | 194 | 139 | 238 | 168 | 249 | 204 |
| DS | 217 | 186 | 164 | 231 | 157 | 245 | 200 |
| Ball | 128 | 126 | 122 | 134 | 127 | 126 | 127 |
| Cube | 131 | 128 | 126 | 130 | 127 | 129 | 128 |
| mean | 173 | 151 | 135 | 179 | 141 | 175 | |

Note a) Experiment is based on running all RP methods, given 10 starting points, with WS=Yes (except for SMC)

Note b) Computation is done on a 2GHz Intel Core Duo computer with 2GB memory

Note c) Time measurements are only given if all 20 iterations are completed and void spaces otherwise.

Note d) For these results the WS method did not result in finding a positive Robustness estimate within 20 iterations

### 5.2.4 Discussion of results

The (confidential) Robust Mixture Design case has practical relevance for Unilever R&D. The described hypothetical Robust Mixture Design case is considered equally relevant. More generally, any product design model that returns product characteristics (y), given explanatory variables ($x$) based on a linear regression model, results in a polyhedral Happy set and for such model all the discussed RP methods are in principle applicable.

The comparison of the RP methods, based on the results in Tables 5.2, 5.3, 5.5 and 5.6 leads to the following conclusions:

1. The Warm Start method is efficient in combination with the MC, N-1MC and DS estimation methods

2. Compression and Decomposition are efficient

3. The SMC and DS method perform above average, with respect to the other Robustness estimation methods

Interestingly, increasing the number of samples does not lead to significantly better results for this case. An explanation is the following. For most of the starting points, the optimum Robustness estimate that is found after 20 iterations, is close to 1. This is not the situation in the other case studies, as discussed in subsequent sections, which show that the number of samples does have a positive effect. It is possible that the sample size increase in the mixture design case, from 100 to 400, is not sufficiently significant to have a positive effect on the performance. More research is required on this issue and it is recommended for future research to study the quality of RP methods when estimating a Robustness close to 1.

## 5.3 Robust Mixture Design for the Exponential case

The Exponential distribution is relevant for reliability optimisation as argued in Section 3.7. There exist interesting reliability optimisation studies, such as by Azaron et al. (2007), for which the RP framework is relevant. However, the case studies investigated in this thesis, are not based on two-sided Exponentially distributed uncontrollable factors. The model (5.15) of the Robust Mixture Design case of Section 5.2 is used to illustrate that the RP methods can in principle solve a relatively large-scale optimisation problem with Exponentially distributed uncontrollable factors. In this study, the Robust Mixture Design case is modified in the following way: In Section 5.2 the elements of random vector $\boldsymbol{v}$ follow a student's $t$-distribution whereas here the uncontrollable factors $\boldsymbol{v}_n$, with $n = 1, .., \mathrm{N}$, are two-sided Exponentially distributed with PDF

$$f(v_n) = \begin{cases} \frac{1}{2}\sqrt{2}e^{-\sqrt{2}v_n} & \text{for} \quad v_n \geq 0 \\ \frac{1}{2}\sqrt{2}e^{\sqrt{2}v_n} & \text{for} \quad v_n < 0 \end{cases} \tag{5.16}$$

It can be shown that $E(\boldsymbol{v}_n) = 0$ and $VAR(\boldsymbol{v}_n) = 1$ with $n = 1, .., \mathrm{N}$. Consequently, the ES method is feasible for this case. However, the DS and Compression method are not applicable for this case, as explained in Chapter 3. The performance indicators $\pi$ (4.42) and $\Pi$ (4.43) for this case, are respectively shown in Table 5.8 and Table 5.9, where the Warm Start method is used.

**Table 5.8**: Performance indicator of combined RP methods for problem (5.15) with two-sided Exponentially distributed $\boldsymbol{v}$

| RP Method $q^{[\kappa]}$ | | | | $\pi^{[\kappa]}$ | p-value $H_a : \mu_{\pi^{[\kappa]}} > \frac{100\%}{20}$ |
|---|---|---|---|---|---|
| Estimator | Comp. | Decom. | sample size | | |
| N-1MC | N.A. | Yes | 400 | 32% | **0.0000** |
| SMC | N.A. | No | 100 | 15% | **0.0043** |
| ES | N.A. | Yes | 400 | 12% | **0.0094** |
| ES | N.A. | No | 400 | 9% | **0.0957** |
| Cube | N.A. | No | N.A. | 7% | **0.2062** |
| N-1MC | N.A. | Yes | 100 | 7% | **0.2410** |
| SMC | N.A. | No | 400 | 6% | **0.3314** |
| SMC | N.A. | Yes | 100 | 6% | 0.3565 |
| ES | N.A. | Yes | 100 | 4% | 0.6596 |
| SMC | N.A. | Yes | 400 | 2% | 0.9798 |

Note a) The results for $\pi > 0$ are given in descending order of $\pi$.

Note b) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note c) WS=Yes (except for SMC) and D=30 strata of L=4 randomly chosen starting points.

**Table 5.9**: Performance indicator of individual RP methods for problem (5.22) with two-sided Exponentially distributed $\boldsymbol{v}$

| RP Method *type* | $\Pi^{[type]}$ | p-value $H_a : \mu_{\Pi^{[type]}} > \frac{100\%}{6}$ | p-value $H_a : \mu_{\Pi^{[type]}} > 50\%$ | p-value $H_a : \mu_{\Pi^{[type]}} < 50\%$ |
|---|---|---|---|---|
| MC | 0.0% | 1.0000 | | |
| SMC | 29.0% | **0.0116** | | |
| N-1MC | 39.0% | **0.0001** | | |
| ES | 25.0% | **0.0465** | | |
| Diamond | 0.0% | 0.0000 | | |
| Cube | 7.0% | 0.9998 | | |
| Decomposition (no MC/SMC) | 72.0% | | **0.0000** | 1.0000 |
| M=400 (no bounding) | 68.0% | | **0.0007** | 0.9993 |

Note a) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note b) WS=Yes (except for SMC) and D=33 strata of L=3 randomly chosen starting points.

**Table 5.10**: Average computation time (sec.) for problem (5.22)

| | Compression | | Decomposition | | Sample size | | |
|---|---|---|---|---|---|---|---|
| | No | Yes$^{d}$ | No | Yes | 100 | 400 | mean |
| MC | | | | | | | |
| SMC | 147 | | 141 | 159 | 138 | 159 | 149 |
| N-1MC | 235 | | 135 | 239 | 178 | 298 | 217 |
| DS | 202 | | 169 | 235 | 159 | 246 | 202 |
| Ball | | | | | | | |
| Cube | | | | | | | |
| mean | 195 | | 149 | 211 | 158 | 234 | |

Note a) Experiment is based on running all RP methods, given 10 starting points, with WS=Yes (except for SMC)

Note b) Computation is done on a 2GHz Intel Core Duo computer with 2GB memory

Note c) Time measurements are only given if all 20 iterations are completed and void spaces otherwise.

Note d) Compression is not applicable for ES

The results show that:

- The efficiency of the SMC, N-1MC and ES method is above average
- The efficiency of the Decomposition method is above average
- Increasing the number of samples is efficient, with respect to the Robustness optimisation result after 20 iterations.
- The computation time is negatively effected by the Decomposition method and by increasing the number of samples.

## 5.4 Optimal Robust Raw Material Cost Planning

As a follow up of the optimal robust product design case, research has been done together with Unilever R&D on making optimal robust decisions that reduce the raw material cost of products. We call this Raw material Cost Planning (RCP). The *original* RCP case is based on confidential data. To sketch the robust RCP problem and the application of the methods for Robustness Programming, the *hypothetical* case of Section 5.2 is extended, as follows.

There are I=11 Raw Materials (RM) used for manufacturing K=2 products. The cost price of the two products are mainly determined by the price of the raw materials. Therefore it is important to find product designs for these two products, that lead to low raw material cost. The price per ton for the raw materials have been recorded for the past 178 weeks and are illustrated on the left in Figure 5.2. These historic prices show considerable volatility and it is fair to assume considerable uncertainty about the raw material prices for future weeks. Besides the cost price aspect, it is important that the product is also satisfying the specifications. The results of the Section 5.2 are used for defining feasible designs as follows. The product design for K products is represented by

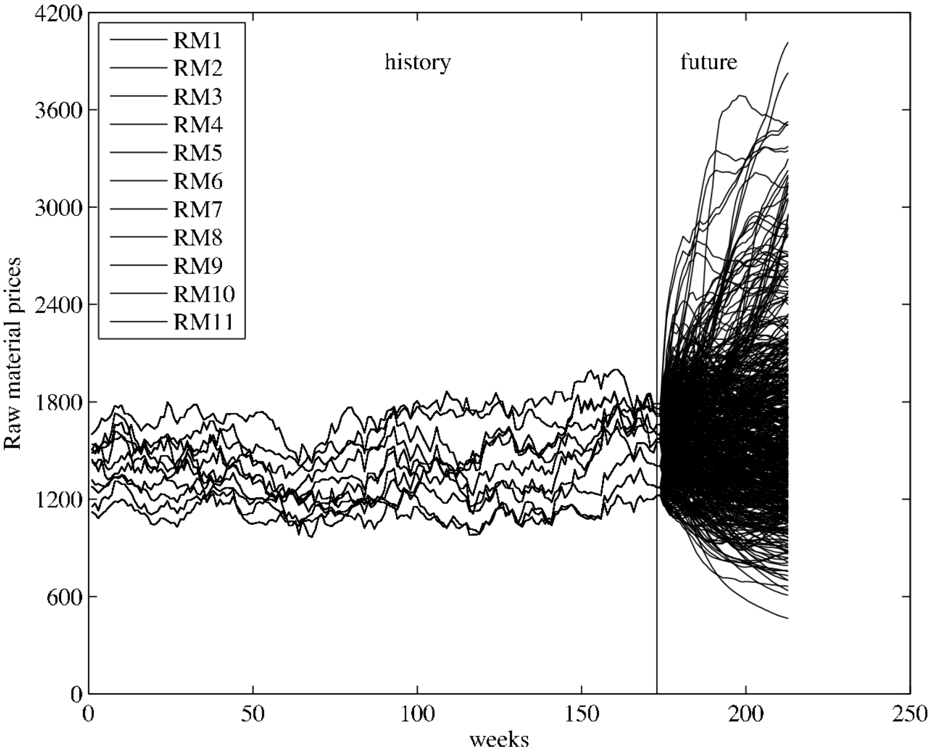


Figure 5.2: Illustration of historic raw material prices and future samples

a I×K matrix

$$X = \begin{bmatrix} \uparrow & & \uparrow & & \uparrow \\ x^{[1]} & \cdots & x^{[k]} & \cdots & x^{[\mathrm{K}]} \\ \downarrow & & \downarrow & & \downarrow \end{bmatrix}$$

where $x^{[k]} \subseteq \mathbb{R}^{\mathrm{I}}$. The set of deterministic feasible designs $\mathbb{X}_k$ for product $k = 1,..,\mathrm{K}$ is defined in (5.12). The Happy set $\mathbb{H}(X)$ is defined in (5.14) and defines the *quality* Robustness

$$R^{[qual]}(X) = \Pr\{\boldsymbol{v} \in \mathbb{H}(X)\} \tag{5.17}$$

In this case the goal is to find a design $X \in \prod_{k=1}^{\mathrm{K}} \mathbb{X}_k$ with at least 95% robustness, with respect to the requirements as given in (5.14). The deterministic specifications and Robustness specifications, can be combined to define the following *feasible* set.

$$\mathbb{F} = \left\{ X \in \mathbb{R}^{\mathrm{I}\times\mathrm{K}} \;\middle|\; \begin{array}{c} X \in \prod_{k=1}^{\mathrm{K}} \mathbb{X}_k \\ R^{[qual]}(X) \geq 0.95 \end{array} \right\} \tag{5.18}$$

Hence, any $X \in \mathbb{F}$ satisfies all product design requirements. What remains, is to find out which $X \in \mathbb{F}$ leads to guaranteed high savings.

The vertical line at number 178 in Figure 5.2 marks the current week. For the past 178 weeks until the current week, the K products have been produced according to design

$$C = \begin{bmatrix} \uparrow & & \uparrow & & \uparrow \\ c^{[1]} & \cdots & c^{[k]} & \cdots & c^{[\mathrm{K}]} \\ \downarrow & & \downarrow & & \downarrow \end{bmatrix} \in \mathbb{F}.$$

The total raw material cost price of the K products, based on design $C$, changed considerably over the past 178 weeks. The objective is to find an alternative product design $X \in \mathbb{F}$, with a high Robustness to save on the current raw material cost price.

The goal is split into two parts: the total saving to be reached and the time span before which to reach the saving. In notation: the total savings goal is $\gamma_1$ and the time span to reach the saving is $\gamma_2$. In this case, the study is limited to only three saving goals of respectively €100,000, €150,000 and €200,000. The timing aspect in the goal (i.e. the second element $\gamma_2$) has the following relevance: If the goal is to save €100,000 in the coming four weeks, then on average €25.000 should be saved per week; if the goal is to save €100,000 in the coming ten weeks, then on average €10.000 should be saved per week and so on.

Figure 5.2 illustrates the price history of 11 RMs for the past 178 weeks and 20 samples per RM, from week 179 until 218. The samples are generated by a Geometric Brownian Motion (GBM) time series model, which is based on the 178 historic observations. The choice for the GBM model is based on the following: Joshi (2003) argues that a GBM model is a standard model in Mathematical Finance for modelling the evolution of stock prices. Additionally, the GBM model plays a central role in Real Options Theory (Johnathan, 2002; Copeland and Antikarov, 2001). From a practical point of view it is important to construct a representative future price sampling model and give statistical evidence of the validity of such model. However, identifying and estimating the most appropriate time series model, based on in-depth statistical analysis, is considered to be outside the scope of this thesis. Therefore, only the generated samples are used and not any other information such as the underlying covariance matrix and probability distribution that generates the GBM samples. This way the RP approach is largely independent of the chosen time series approach.

Let $\boldsymbol{v}$ be a random $11 \times 40$ matrix, representing the random future prices per ton for each RM for the weeks 179 until 218, for which samples are generated with the GBM model. Let the weekly production volume of product $k$ be $q_k$ (in 1000Kg units) for $k = 1, .., \mathrm{K}$. The current week is 178. Let $\tau = 1$ correspond to the first future week, namely 179; $\tau = 2$ corresponds to future week 180; etcetera. The total saving over future weeks until moment in time $\gamma_2$, for a mixture design $X \in \mathbb{F}$ with respect to the cost of the mixture design $C \in \mathbb{F}$, is

$$u^{[\gamma_2]}(x, v) = \sum_{i=1}^{\mathrm{I}} \sum_{k=1}^{\mathrm{K}} q_k \left(\mathrm{C}_{i,k} - X_{i,k}\right) \sum_{\tau=1}^{\gamma_2} v_{\tau,i} \tag{5.19}$$

given future RM prices $v \in \mathbb{V}$. We are interested in the Robustness of the total savings (5.19) reaching goal $\gamma_1$ before moment in time $\gamma_2$:

$$R^{[\gamma]}(x) = \Pr\left\{\boldsymbol{v} \in \mathbb{H}^{[\gamma]}(x)\right\}, \text{ where } \mathbb{H}^{[\gamma]}(x) = \left\{v \in \mathbb{R}^{\mathrm{I}\times 40} | \gamma_1 \leq u^{[\gamma_2]}(x, v)\right\} \tag{5.20}$$

The corresponding Robustness Programming Set (RPS) is

$$\mathbb{P} = \left\{ \left( \begin{array}{c} \gamma \\ X^* \\ R^* \end{array} \right) \middle| R^* = \max_{X \in \mathbb{F}} R^{[\gamma]}(X), R^* = R^{[\gamma]}\left(X^*\right) \right\} \tag{5.21}$$

where we vary $\gamma_1 \in \{100000, 150000, 200000\}$ and $\gamma_2 = 1, .., 40$.

### 5.4.1 Results

Methods for Robustness Programming are used to find RPS elements. Figure 5.3 illustrates the optimal Robustness for given goals $\gamma$. The corresponding optimal designs $X^*$ have a large dimension, i.e. K×I×40×3, which in this case corresponds to 120 matrices (one design matrix $X$ for each illustrated point in Figure 5.3) each consisting of 22 elements. The Matlab FMINCON solver has been used to solve the constrained optimisation problem

$$R^* = \max_{X \in \mathbb{F}} R^{[\gamma]}(X) \tag{5.22}$$

which defined the RPS elements.

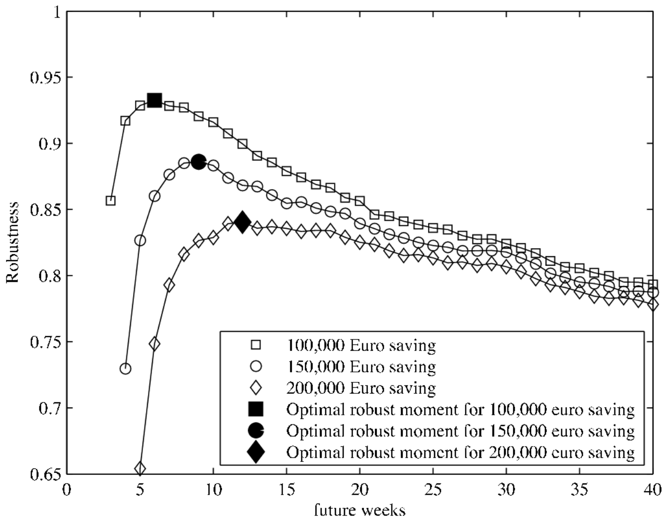


Figure 5.3: Illustration of RPS $\mathbb{P}$ (5.21), where ◯, ◊ and □ mark computed $R^*$ for corresponing values of $\gamma$. The solid lines are interpolations between future weeks saving moments $\gamma_2$, for a given saving goal $\gamma_1 \in \{100000, 150000, 200000\}$

The RCP case is an extension of the Multiple Product Design (MPD) case of Section 5.2. Consequently, the complexity of the RCP problem (e.g. the number of constraints and random variables) is larger and results in longer computation times that make the experiments too intractable. With the following approach the computation time could be halved: The same methods as in Section 5.2 can be used to verify if $X \in \mathbb{F}$ and solving the MPD problem of Section 5.2 is an approach to *find* an $X \in \mathbb{F}$. Hence, the optimal designs found for the MPD problem can be used as starting points for the RCP problem. This approach is characterised by two phases:

- Phase-1: Solve $f_{\kappa}^{\text{RPSolve}}\left(X^{[l]}\right) = \left(\tilde{R}\left(X^{[l,\kappa]*}\right)\right)$ for problem (5.15).
- Phase-2: Use the optimal design of phase-1 as a starting point, i.e. $X^{[l]} := X^{[l,\kappa]*}$ and Solve $f_{\kappa}^{\text{RPSolve}}\left(X^{[l]}\right) = \left(\tilde{R}\left(X^{[l,\kappa]*}\right)\right)$ for problem (5.22).

The same RP methods $q^{[\kappa]}$, $\kappa = 1,..,$K as in Section 5.2 are applicable to estimate the design robustness (5.17). However, to estimate the cost price saving Robustness $R^{[\gamma]}(x)$, only the MC and SMC methods are applicable[4]. The following RP method assignment rule is used for solving problem (5.22): If the design Robustness $R^{[d]}(x)$ is estimated with method $\text{q}^{[\kappa]}$=($\text{q}_\text{E}$,$\text{q}_\text{C}$,$\text{q}_\text{D}$,$\text{q}_\text{W}$,$\text{q}_\text{S}$), then the cost price saving Robustness $R^{[\gamma]}(x)$ is estimated with the MC method if $\text{q}_\text{E}$=MC and with SMC for all other values of $\text{q}_\text{E}$.

The performance indicators $\pi$ (4.42) and $\Pi$ (4.43) for the RCP case, are respectively shown in Table 5.11 and Table 5.12, where the Warm Start method is used.

**Table 5.11**: Performance indicator of combined RP methods for problem (5.22)

| RP Method $q^{[\kappa]}$ | | | | $\pi^{[\kappa]}$ | p-value<br>$H_a : \mu_{\pi^{[\kappa]}} > \frac{100\%}{48}$ |
|---|---|---|---|---|---|
| Estimator | Comp. | Decom. | sample size | | |
| DS | No | Yes | 400 | 10% | **0.0027** |
| DS | Yes | Yes | 100 | 10% | **0.0054** |
| N-1MC | Yes | Yes | 400 | 9% | **0.0294** |
| DS | Yes | Yes | 400 | 9% | **0.0164** |
| DS | No | No | 400 | 8% | **0.0090** |
| Ball | Yes | Yes | N.A. | 7% | **0.0190** |
| N-1MC | No | Yes | 100 | 6% | **0.0470** |
| N-1MC | No | Yes | 400 | 6% | **0.0388** |
| N-1MC | Yes | Yes | 100 | 6% | **0.0470** |
| Cube | Yes | Yes | N.A. | 5% | 0.0769 |
| DS | No | Yes | 100 | 4% | 0.1470 |
| SMC | Yes | Yes | 100 | 4% | 0.1646 |
| SMC | No | No | 400 | 3% | 0.2700 |
| Ball | Yes | No | N.A. | 3% | 0.3310 |
| DS | No | No | 100 | 3% | 0.4692 |
| SMC | No | Yes | 100 | 3% | 0.4692 |
| Cube | Yes | No | N.A. | 2% | 0.7419 |
| SMC | Yes | Yes | 400 | 1% | 0.9701 |

Note a) The results for $\pi > 0$ are given in descending order of $\pi$.

Note b) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note c) WS=Yes (except for SMC) and D=30 strata of L=4 randomly chosen starting points.

Note d) The optimal designs $X^{[l,\kappa]*}$ of the MPD case of Section 5.2 are used as starting points $X^{[l]}$, $l = 1,..,$L

[4] The Compression, N-1MC, DS, ES, Diamond, Ball or Cube method cannot be applied, because the distribution of the random future raw material prices is not known. The Decomposition method is not relevant, because $u^{[\gamma_2]}$ has a one dimensional range

**Table 5.12**: Performance indicator of individual RP methods for problem (5.22)

| RP Method *type* | $\Pi^{[type]}$ | p-value $H_a : \mu_{\Pi^{[type]}} > \frac{100\%}{6}$ | p-value $H_a : \mu_{\Pi^{[type]}} > 50\%$ | p-value $H_a : \mu_{\Pi^{[type]}} < 50\%$ |
|---|---|---|---|---|
| MC | 0.0% | 1.0000 | | |
| SMC | 11.3% | 0.9736 | | |
| N-1MC | 27.0% | **0.0100** | | |
| DS | 44.3% | **0.0000** | | |
| Ball | 10.4% | 0.9870 | | |
| Cube | 7.0% | 1.0000 | | |
| Compression (no MC/SMC) | 58.3% | | **0.0228** | 0.9772 |
| Decomposition (no MC/SMC) | 80.0% | | **0.0000** | 1.0000 |
| M=400 (no bounding) | 57.4% | | 0.0711 | 0.9289 |

Note a) The methods in the rows with the **bold** p-values are the best performing methods beyond reasonable doubt.

Note b) WS=Yes (except for SMC) and D=30 strata of L=4 randomly chosen starting points.

Note c) The optimal designs $X^{[l,\kappa]*}$ of the MPD case of Section 5.2 are used as starting points $X^{[l]}$, $l = 1, .., \mathrm{L}$

**Table 5.13**: Average computation time (sec.) for problem (5.22)

| | Compression | | Decomposition | | Sample size | | |
|---|---|---|---|---|---|---|---|
| | No | Yes | No | Yes | 100 | 400 | mean |
| MC | | $163^{d}$ | $163^{d}$ | | $152^{d}$ | $164^{d}$ | 160 |
| SMC | 164 | 173 | 146 | 176 | 147 | 189 | 166 |
| N-1MC | 262 | 247 | 172 | 281 | 181 | 310 | 242 |
| DS | 254 | 238 | 198 | 285 | 187 | 303 | 244 |
| Ball | | 136 | 131 | 141 | 139 | 134 | 136 |
| Cube | | 133 | 127 | 137 | 130 | 135 | 132 |
| mean | 226 | 182 | 156 | 204 | 156 | 206 | |

Note a) Experiment is based on running all RP methods, given 10 starting points, with WS=Yes (except for SMC)

Note b) Computation is done on a 2GHz Intel Core Duo computer with 2GB memory

Note c) Time measurements are only given if all 20 iterations are completed and void spaces otherwise.

Note d) For these results the WS method did not result in finding a positive Robustness estimate within 20 iterations

### 5.4.2 Discussion of results

Unilever R&D is interested in the idea of decision support information given by the RPS set in combination with graphical information such as Figure 5.3. Therefore, RP methods are relevant because these can be used to identify elements in the RPS set and give the information to make a graph such as in Figure 5.3. The following can be concluded on the performance of the RP methods.

1. The Compression and Decomposition method are efficient
2. The SMC method is necessary to solve this case, since MC fails and the other estimation methods cannot be used for estimating the price Robustness $R^{[\gamma]}(x)$
3. The N-1MC and DS method perform above average, with respect to the other Robustness estimation methods.

4. Decomposition and the sample size both have a negative effect on the computation time

5. Compression has a positive effect on the computation time

## 5.5 Coalition Robustness in Cartel Formation Game

Game theory models on coalition formation deal with a group of decision makers deciding to agree on cooperation, because it increases their benefit compared to a situation of no cooperation. Stable coalitions have been described in this perspective by among others Yi (1997). A coalition of players is considered stable, if no player inside the coalition has the incentive to leave the coalition and no player outside the coalition has the incentive to join the coalition. More recently, the idea has been applied to get a feeling for incentives to form international climate agreements on reduction of the emission of greenhouse gasses (the so-called Kyoto discussions) in some empirical studies. Eyckmans and Finus (2003) study the world divided into 6 regions and the tendency to cooperate between the regions using estimated payoff models. Their study is elaborated and extended by Finus et al. (2003) using new estimates of the payoff function for 12 world regions and algorithms developed in cooperation with Wageningen University and Hagen University in a research group called STACO. The final outcome of the analysis is whether a coalition is stable.

The case described in this section is based on an article by Olieman and Hendrix (2006), where the concept of coalition stability is extended to the concept of Coalition Robustness, defined as the probability for coalition stability, the so-called *Stability Likelihood*. Coalition Robustness is based on the assumption that a probability space can be defined, which models the uncertainty about estimated model parameters in the underlying payoff functions. In Olieman and Hendrix (2006) the application of the combination of MC and DS Robustness estimation methods on the STACO case is studied as will be elaborated further in this section.

The Coalition Robustness in the STACO case will be discussed in the following order: A formal description of a coalition formation game and the definition of Coalition Robustness is given in Section 5.5.1. In Section 5.5.2 the STACO model is defined. In Section 5.5.3 the performance of the Robustness estimation methods is discussed, in the context of the STACO model.

### 5.5.1 Cartel Coalition Stability

**Coalition Formation Game**

The terminology and mathematical formalisation are based on Yi (1997), Finus et al. (2003) and Eyckmans and Finus (2003). Consider a set of I players $\mathbb{I} = \{1, \cdots, \mathrm{I}\}$, where each player $i \in \mathbb{I}$ expects a payoff, based on a payoff model $\pi_i(q, v)$, as a consequence of player strategies $q \in \mathbb{R}^{\mathrm{I}}$ and model parameters $v \in \mathbb{R}^{\mathrm{N}}$. The payoff function is elaborated on in (5.31). The payoff functions are given as polynomials where the elements of the strategy $q$ are the independent variables and the elements of parameter vector $v$ are the coefficients of these polynomials. The main idea in the Coalition Robustness case, is to study the situation that $\boldsymbol{v}$ is a random vector. The coalition formation game is introduced assuming a constant $v$, before studying the Coalition Robustness for the situation that $\boldsymbol{v}$ is a random vector.

Besides the strategy choice $q$, players are also considered to make a choice about forming a coalition. Focus is on *cartel* coalition formation, which means there is only one coalition at the same time. The idea of multiple coalition formation games is studied in detail in Sáiz et al. (2006). In principle, the ideas given in this section can be extended to multiple coalition formation games.

The choice of players to form a *cartel coalition* $\mathbb{C} \subseteq \mathbb{I}$, is modelled by a binary vector $c \in \{0,1\}^{\mathrm{I}}$, meaning player $i$ can either join ($c_i = 1$) or not join ($c_i = 0$) the coalition. Vector $c$ is called the *coalition strategy* and defines the *cartel coalition*:

$$\mathbb{C}(c) = \{i \in \mathbb{I} | c_i = 1\} \tag{5.23}$$

To analyse and predict rational strategies, the cartel coalition formation is modelled as a *two-stage game*. In the *first stage*, players decide on their coalition strategy ($c$); in the *second stage* players decide what their strategy ($q$) would be, if coalition strategy $c$ would be the case.

The concept of *Nash Equilibrium* is used to define a stable strategy: Strategy $c^*$ is a Nash Equilibrium, if no player $i$ has the incentive to deviate from its strategy $c_i^*$, given that the other players $j \neq i$ do not deviate from their strategies $c_j^*$, see e.g. Rasmusen (1994). In other words: Strategy $c^*$ is a Nash Equilibrium if no player has the incentive for *individual strategy deviation.*

### Stable strategies at the second stage

The payoff functions $\pi_i(q,v)$ with $i = 1,.,,\mathrm{I}$ are assumed to be continuous in both $q$ and $v$. At the second stage, coalition members are considered to maximise the *aggregate payoff* $\sum\limits_{i\in\mathbb{C}(c)} \pi_i(q,v)$ of the coalition, while the other players maximise their *individual payoff* $\pi_i(q,v)$ for $i \in \mathbb{I} \setminus \mathbb{C}(c)$. The objective of player $i$ at stage 2, in connection to some coalition strategy $c$, is to maximise $\Pi_i$ over $q_i$, with:

$$\Pi_i(q,v,c) = \begin{cases} \sum\limits_{i\in\mathbb{C}(c)} \pi_i(q,v) & \text{if } i \in \mathbb{C}(c) \\ \pi_i(q,v) & \text{if } i \notin \mathbb{C}(c) \end{cases}$$

The vector $\Pi$ is called the *aggregate payoff vector.* Let $\mathbb{E}_i \subset \mathbb{R}$, with $i = 1,..,\mathrm{I}$ be a compact set representing the feasible set of strategy $q_i$ of individual player $i$, such that over all players $q \in \mathbb{E} = \prod\limits_{i=1}^{\mathrm{I}} \mathbb{E}_i$.

To define the set of stable strategies at the second stage, the Nash Equilibrium condition is used. Possibly, the Nash Equilibrium condition might hold for infinitely many strategies in $\mathbb{E}$ (see for example Osborne and Rubinstein, 1994). The set of *stable strategies* at the second stage is defined by:

$$\mathbb{S}(c,v) = \{q^* \in \mathbb{E} | \forall_i \forall_{\{q|q_j = q_j^*, q_i \in \mathbb{E}_i, j \neq i\}} : \Pi_i(q^*,v,c) \geq \Pi_i(q,v,c)\} \tag{5.24}$$

It can be shown that the set $\mathbb{S}(c, v)$ is compact, since $\mathbb{E}$ is compact and $\Pi_i(q, v, c)$ with $i = 1, ..,$I is continuous in $q$ and $v$, for a given coalition $c$.

### Coalition stability condition

At the first stage, the strategy space of player $i$ is to either choose to join ($c_i = 1$) or not to join ($c_i = 0$) the cartel coalition. In analogy to the concepts in the paper of Yi (1997), a coalition strategy $c$ is defined stable, if this coalition strategy is in Nash Equilibrium.

So-called *neighbouring* coalition strategies $nc(c, i)$ for $i = 1, ...,$I are introduced, to represent the Nash Equilibrium characteristic of individual strategy deviation: The vector $nc(c, i)$ represents the situation in which only player $i$ chooses the alternative strategy with respect to the strategy in $c$, while the other players $j \neq i$ do not change their strategy. For example, let I $= 4$. Consider a coalition strategy vector $c = [1, 0, 1, 1]^{\intercal}$. The neighbouring coalition strategies are:

$$\begin{aligned} nc(c, 1) &= [0, 0, 1, 1]^{\intercal} \\ nc(c, 2) &= [1, 1, 1, 1]^{\intercal} \\ nc(c, 3) &= [1, 0, 0, 1]^{\intercal} \\ nc(c, 4) &= [1, 0, 1, 0]^{\intercal} \end{aligned} \tag{5.25}$$

More formally, the neighbouring coalition strategies of $c$ are defined as:

$$nc_j(c, i) = \begin{cases} c_j & \text{if } j \neq i \\ 1 - c_j & \text{if } j = i \end{cases}$$

To test whether player $i$ has an incentive to deviate from $c_i$, one should compare the payoff of player $i$ in coalition strategy $c$, with the payoff of player $i$ in the neighbouring coalition strategy $nc(c, i)$.

Recall that at stage 2, the set $\mathbb{S}(\cdot)$ may contain more than one element. In that situation we have no argument to determine which of these strategies will be chosen, since they are all equilibria. As a consequence, we cannot determine the payoff values in connection to strategy $c$. However, since the set is compact, we can conclude about the upper- and lower bounds, which leads to a *sufficient* condition for coalition stability.

A coalition strategy $c$ is defined to be stable if: *For all players $i$, the worst stage 2 payoff strategy in connection to coalition strategy $c$, is better than or equal to the best stage 2 payoff strategy in connection to $nc(c, i)$.* More formally, if the worst case payoff difference

$$u_i(c, v) = \min_{q \in \mathbb{S}(c,v)} \pi_i(q, v) - \max_{q \in \mathbb{S}(nc(c,i),v)} \pi_i(q, v) \tag{5.26}$$

is non-negative for all players

$$\forall_{i \in \mathbb{I}} : u_i(c, v) \geq 0 \tag{5.27}$$

then the coalition strategy $c$ is defined stable. The indicator function

$$I(c, v) = \begin{cases} 1 & \text{if } \forall_{i \in \mathbb{I}} : u_i(c, v) \geq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{5.28}$$

indicates the stability, given $c$ and $v$. Notice that $u(c, v)$ is not continuous in $c$. The objective in this case study is to estimate the Robustness for a finite number of $c$ values, rather than optimising the Robustness over $c$. In this situation, the continuity aspect of $u(c, v)$ with respect to $c$, is not relevant and the RP framework can be used for estimating Robustness.

**Coalition Robustness**

If there is certainty about the model parameter vector $v$, then the payoff model $\pi(q, v)$ is assumed to be *correct*. However, this is often not the case in practice, because $v$ is estimated based on empirical data or based on expert judgement. Focus is on the situation where there is uncertainty about the value of $v$ and this uncertainty is modelled by a stochastic vector $\boldsymbol{v}$, with probability space $(\mathbb{V}, \mathcal{V}, \Pr)$ and a *probability density function* $f(v)$. The measurable set of model parameter values, for which coalition strategy $c$ is stable, is defined as

$$\mathbb{H}(c) = \{v \in \mathbb{R}^n | I(x, v) = 1\} \tag{5.29}$$

This leads to the definition of *Coalition Robustness* $R(c)$ of coalition strategy $c$:

$$R(c) = \Pr\{\boldsymbol{v} \in \mathbb{H}(c)\} = \int_{\mathbb{H}(c)} f(v)dv \tag{5.30}$$

From the conceptual point of view, the Coalition Robustness of strategy $c$, is the probability that strategy $c$ defines a Nash Equilibrium.

In a study of Finus et al. (2003), the authors claim some coalition $c$ to be stable, assuming correct estimation of the model parameters $v$. However, dropping the assumption of "knowing" the value of $v$ and replacing it by the assumption that we know the probability distribution of $v$, implies the Coalition Robustness to be the *probability that the coalition stability claim is correct.* This information is considered relevant in the context of uncertainty analysis of game theoretical coalition formation models.

The concept of Coalition Robustness and the Robustness estimation algorithms described in Chapter 3, have been used to estimate Coalition Robustness in the so-called STACO case. A detailed description of the STACO model can be found in Finus et al. (2003). The following section gives an illustration of the STACO model.

### 5.5.2 STACO model

The STACO project investigates the formation and stability of international climate agreements. The basic structure of the STACO models consists of interacting regions that (i) choose to join an international climate agreement or not; and (ii) choose their optimal climate policy given the coalition formed. The regions are characterised by their abatement costs and damage functions and linked via global climate change and the possibility to establish an international agreement. The purpose of the STACO model, is to find stable coalitions of players, in the context of cost-efficient reduction of $CO_2$ emissions. In the STACO model, the set of world players are simplified into a set of I = 12 regions (see A.11). The STACO model uses a payoff model $\pi(\cdot)$ to model the payoff as function of $CO_2$ emission reductions $q$.

### Payoff regression model

The payoff model used in Finus et al. (2003), considers the *benefits* ( i.e. prevented damages) to be related to the total $CO_2$ emission reduction $(\sum_{i\in\mathbb{I}} q_i)$, whereas the *cost* is considered to only depend on the individual $CO_2$ emission reduction $(q_i)$:

$$\pi(q,v) = \begin{cases} \pi_1(q,v) = \beta_{1,1}\sum_{j=1}^{12} q_j & -\frac{1}{2}\beta_{1,2}q_1^2 - \frac{1}{3}\beta_{1,3}q_1^3 \\ & \cdots \\ \pi_i(q,v) = \beta_{i,1}\sum_{j=1}^{12} q_j & -\frac{1}{2}\beta_{i,2}q_i^2 - \frac{1}{3}\beta_{i,3}q_i^3 \\ & \cdots \\ \pi_{12}(q,v) = \underbrace{\beta_{12,1}\sum_{j=1}^{12} q_j}_{\text{benefit}} & \underbrace{-\frac{1}{2}\beta_{12,2}q_{12}^2 - \frac{1}{3}\beta_{12,3}q_{12}^3}_{\text{cost}} \end{cases}, \tag{5.31}$$

The first argument in $\pi(\cdot)$ is the vector representing the 12 regions $CO_2$ reduction strategies: $q = [q_1, \cdots, q_i, \cdots, q_{12}]^\intercal$. The second argument are the model parameters, which are *stacked* in a vector to shorten notation: $v = (\beta_{1,1}, \beta_{1,2}, \beta_{1,3}, \cdots, \beta_{12,1}, \beta_{12,2}, \beta_{12,3})^\intercal$. The model parameter vector $v$ has N = 36 elements.

### RP framework and the coalition stability check

Results of this case study can be found in Olieman and Hendrix (2006) and Dellink et al. (2007). The RP methods under investigation are DS and MC methods. This section summarises the main results of this case study.

Appendix A.12 shows that (5.24), based on the STACO payoff model (5.31), for any cartel coalition strategy $c \in \{0,1\}^{\mathrm{I}}$ and continuous random model parameter vector $\boldsymbol{v}$, almost surely (a.s.) defines a unique efficient $CO_2$ reduction strategy $q^*$. Consequently, the Nash equilibrium $CO_2$ reduction strategies in the STACO case are a.s. uniquely defined by the coalition strategies via $\{q^*\} = \mathbb{S}(c,v)$ and $\{q^{**}\} = \mathbb{S}(nc(c,i),v)$ and the payoff difference function (5.26) can be simplified to:

$$u_i(c,v) = \pi_i(q^*,v) - \pi_i(q^{**},v) \tag{5.32}$$

Moreover, Appendix A.12 shows that (5.32) is continuous in each point $v$, given strategy $c$, where for all $i = 1,..,12$

$$\beta_{i,2}^2 + 4\beta_{i,3}\gamma_i(c,v) \neq 0 \tag{5.33}$$

$$\beta_{i,2}^2 + 4\beta_{i,3}\gamma_i(nc(c,i),v) \neq 0 \tag{5.34}$$

$$\beta_{i,2} \neq 0 \tag{5.35}$$

$$\beta_{i,3} \neq 0 \tag{5.36}$$

with

$$\gamma_i(c,v) = \left( c_i \sum_{j \in \mathbb{C}(c)\setminus\{i\}} \beta_{j,1} \right) + \beta_{i,1}$$
$$v = (\beta_{1,1}, \beta_{1,2}, \beta_{1,3}, \cdots, \beta_{12,1}, \beta_{12,2}, \beta_{12,3})^{\intercal}$$

From a theoretical point of view, the discontinuity of the payoff difference function, does not fit into the RP framework. However, since the discontinuity conditions are known, the problem can be isolated and dealt with separately. A practical approach is to redefine the Happy set as

$$\mathbb{H}(c) = \left\{ v \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} u_i(c,v) & \geq 0 \\ |\beta_{i,2}^2 + 4\beta_{i,3}\gamma_i(c,v)| & \geq \delta \\ |\beta_{i,2}^2 + 4\beta_{i,3}\gamma_i(nc(c,i),v)| & \geq \delta \\ |\beta_{i,2}| & \geq \delta \\ |\beta_{i,3}| & \geq \delta \end{array} \quad \text{for } i = 1,..,12 \right\}$$

with

$$\gamma_i(c,v) = \left( c_i \sum_{j \in \mathbb{C}(c)\setminus\{i\}} \beta_{j,1} \right) + \beta_{i,1}$$

and set the value of $\delta$ small enough[5] such that

$$\Pr \left\{ v \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} |\beta_{i,2}^2 + 4\beta_{i,3}\gamma_i(c,v)| & \leq \delta \\ |\beta_{i,2}^2 + 4\beta_{i,3}\gamma_i(nc(c,i),v)| & \leq \delta \\ |\beta_{i,2}| & \leq \delta \\ |\beta_{i,3}| & \leq \delta \end{array} \quad \text{for } i = 1,..,12 \right\} \approx 0$$

This Happy set definition fits the RP framework with respect to $v$.

**Stochastic model of model parameter values**

The probability space defined in (Olieman and Hendrix, 2006) is used for the numerical illustration of Coalition Robustness estimation, of which the results are given in the next section. All uncertain model parameters are considered independent normally distributed with mean values such as defined in Table A.2 and a standard deviation that is 5% of these mean values.

## 5.5.3 Results

In the paper of Finus e.a. Finus et al. (2003), a stable coalition between the European Community (EEC) and Japan was found in three different scenarios, which are respectively called the "120%", "200%" and "300%" scenario. These three scenarios refer to three different assumptions related to the mean value in the benefit term of the payoff

[5] A setting for $\delta$ is the numerical precision of the computer.

model. (see table A.2). The Coalition Robustness estimates in this illustration do not reflect the actual uncertainty analysis of the studies in (Finus et al., 2003) and (Dellink et al., 2007). The target standard error of the EEC-Japan Coalition Robustness estimate was set to 0.02. The estimator function $\overset{\text{mc}}{\boldsymbol{R}}$ as defined in (3.1) is used to compute the MC estimate, given M samples $v^{[\mathrm{M}]}$ of $\boldsymbol{v}$. The estimator $\overset{\text{ds}}{\boldsymbol{R}}(c) = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} \overset{\text{ds}}{r}(\boldsymbol{s}^{[m]}, c)$ as defined in (3.67), is used to compute the DS estimate, given M samples $s^{[\mathrm{M}]}$ of $\boldsymbol{s}$, which are uniformly distributed random points on the unit sphere.

The Happy set $\mathbb{H}(c)$ in the STACO case is not polyhedral. The boundary of the Happy set is defined by non-linear equations (which can be concluded from the elaboration in Appendix A.12). Consequently, the ray intersection distances as given in Definition 3.20, cannot be determined as efficiently as in the case of polyhedral Happy sets. Instead, these ray intersection distances are determined numerically with the Matlab Secant method (called FZERO and explained by Van Loan (2000)). Furthermore, computing $\overset{\text{ds}}{r}(s^{[m]}, c)$ is based on evaluating the Chi square CDF with the Matlab CHI2CDF function, which is computational intensive.

The results were computed with Matlab® version 5. This version includes the function called FLOPS, that returns the number of floating point operations executed, to compute the results. Table 5.14 gives the performance results of the MC method versus the DS method.

**Table 5.14**: Coalition Robustness estimation performance results

| | $\hat{R}(c) = \overset{\text{mc}}{R}(c)$ | | | $\hat{R}(c) = \overset{\text{ds}}{R}(c)$ | | |
|---|---|---|---|---|---|---|
| scenario | 120% | 200% | 300% | 120% | 200% | 300% |
| $\hat{R}(c)$ | 0.49 | 0.82 | 0.91 | 0.51 | 0.83 | 0.92 |
| samples | 617 | 356 | 191 | 615 | 308 | 147 |
| FLOPS | 5,452,424 | 3,099,509 | 1,647,179 | 26,570,249 | 28,162,074 | 12,509,752 |
| FLOPS/sample | 8,836 | 8,706 | 8,623 | 43,203 | 91,435 | 85,100 |

### 5.5.4 Discussion of Results

Both Robustness estimation approaches give comparable results regarding the estimated Coalition Robustness, i.e. the differences between the estimated Coalition Robustness is smaller than the target standard error (0.02). However, the observed number of samples and FLOPS that are necessary to compute these results, lead to the following conclusions.

- The DS method uses less samples than the MC method, in particular in the situation of estimating higher Robustness values (e.g. the 300% scenario). These observations are supported by Theorem 3.6 and lead to the conclusion that the DS method is more efficient than the MC method in the STACO case and given coalition $c$, in terms of the required number of samples.

- In the 120% scenario the difference between the MC and DS method with respect to the required samples, is the smallest. An explanation is given by Olieman and Hendrix (2006). It is shown that for 312 samples the estimate function $\overset{\text{ds}}{r}(s^{[m]}, c) \geq 0.9$ and for 303 samples $\overset{\text{ds}}{r}(s^{[m]}, c) \leq 0.1$ of in total 615 samples. This means that the Robustness estimate per sample is either close to 1 or close to 0. This characteristic corresponds to the idea of a Happy set that is DS-Radial-Shaped as in Theorem 3.7 and explains why the DS and MC method use a similar number of samples.

- The DS method requires more FLOPS to come to the same result as the MC method, for each scenario. Furthermore, the observed average FLOPS per sample are different for the scenarios for the DS method. From a computational point of view, this leads to the conclusion that the MC method is more efficient than the DS method and has more predictable behaviour. It is likely that the inefficiency of the DS method is due to the computational burden of the FZERO approximation of the ray intersection distances and the CHI2CDF function evaluation to compute the Robustness estimate, given sample $s^{[m]}$.

## 5.6 Concluding remarks

In this chapter the case studies of this research have been discussed. The first case is described in Section 5.2 and deals with optimal robust mixture design of a food product. The second case is given in 5.4 and deals with optimal robust raw material cost planning in a food production context. The last case is outlined in Section 5.5 and deals with coalition robustness in a two-stage cartel game, in the context of $CO_2$ reduction strategies in climate agreements. The cases were used to assess the performance of the RP methods of Chapters 3 and 4. In the next chapter, the conclusions and recommendations of this research are formulated.

# Chapter 6
# Conclusions and Recommendations

## 6.1 Introduction

The main objective of this study is to define a systematic framework for Robustness Programming and to develop methods for solving Robustness Programming problems. The RP framework should enable the identification of applicable RP methods, when given an RP problem with specific properties. Existing Robustness estimation methods are improved and generalised and new estimation methods are designed. The effectiveness and efficiency of the RP methods are investigated and methods to improve these are defined. Finally, the RP framework and RP methods are used to solve practical cases. Conclusions about the RP framework, the mathematical properties of RP methods and numerical results are respectively given in 6.2, 6.3 and 6.4. Finally, recommendations for future research are given in 6.5.

## 6.2 Robustness Programming framework

Robustness of an object is defined as the probability that an object will have properties as required. Robustness programming is the search for designs that have a Robustness as big as possible. Robustness Programming is based on Stochastic Programming, Probability theory and Mathematical Programming. The methods for Robustness Programming presented in this thesis, extend the set of methods for solving practical decision problems under uncertainty. A Stochastic Programming problem, that is defined by chance constraint functions and a probabilistic objective function, can be translated to an RP problem by defining the corresponding uncertain object properties function $u(x, v)$. Three cases were studied to show the practical relevance of Robustness Programming. A mixture design case and a material cost planning case are studied for Unilever R&D and a game theoretical coalition formation case is studied as part of the research program called STACO the Wageningen University. The following can be concluded about the RP Framework, with respect to the first and second research question, as given in Section 1.6:

1. The RP framework, as defined in Chapter 2, gives a systematic notation to describe RP methods and RP problems. The controllable factors $x$, the uncontrollable random factors $\boldsymbol{v}$, the Happy set $\mathbb{H}(x)$ and the Robustness definition (2.4) are the generic components for defining an RP problem. The continuity of $u(x, v)$ with respect to $x$, is not relevant when computing the Robustness for a given $x$, but is relevant for optimisation of Robustness using standard NLP methods.

2. The Robustness Programming Framework makes it possible to identify appropriate RP methods, based on the mathematical properties of the RP problem as summarised in Table 6.1

## 6.3 Mathematical Properties of RP methods

The third research question deals with identifying mathematical properties of RP methods, that give information about the quality of RP methods, relevant for Robustness estimation and Robustness optimisation. In this respect, the effectiveness, efficiency and applicability of RP methods are relevant for comparing the quality of RP methods. The following RP methods were developed and investigated:

- *Robustness estimation methods based on sampling*: MC, SMC, N-1MC, DS and ES method
- *Robustness estimation methods based on bounding*: Diamond, Ball and Cube method
- *Computation methods*: Compression, Decomposition and Warm Start method

In this list, the MC method is considered to be one of the most common statistical methods to estimate a probability. The MC methods is used as a reference method for comparing the alternative RP methods. The DS method is a generalisation of the methods developed by István Deák (2000, 2003). All other methods were developed during this research.

### Effectiveness

In general, Robustness $R(x)$ cannot be computed directly and is estimated instead. The objective is to estimate $R(x)$ effectively. An RP method is defined effective if the RP estimation method is unbiased, since that would imply $R(x)$ can be estimated arbitrarily accurately, given sufficient samples.

The MC, N-1MC, DS and ES method are effective estimation methods because they are all based on unbiased Robustness estimators. The SMC method deviates at most $\frac{1}{2\mathrm{M}}$ from the MC estimate, where M is the number of samples. This means that the SMC estimate can be made arbitrarily close to the unbiased MC estimate, by choosing M large enough.

The effectiveness of the Diamond, Ball and Cube bounding methods depends on the underlying RP problem. To identify the most effective bound involves solving Mathematical Programming (MP) problem (3.98) or (3.100). Only if the global solution is found for the corresponding MP problems, the bound is correct[1] and optimally effective.

The Compression and Decomposition method do not change the unbiasedness of estimation methods. However, Example 4.2 illustrates how compression leads to more effective bounds and Corollary 4.5 and Corollary 4.6 identify the conditions, such that decomposition leads to strictly more effective bounds.

### Efficiency

The efficiency of Robustness estimation methods, that are based on sampling, is measured as the standard error of the estimator given M=1 sample. It has been found out that the

[1] If the MP solution is a local solution instead of the global solution, then in the case of lower bounding it is possible that the bounding set $\mathbb{B}^{[p]}$ is not a subset of the Happy set and in the case of upper bounding it is possible that the bounding set is not a superset of the Happy set. This is illustrated in Figure 3.22

N-1MC, DS and ES methods are strictly more efficient than the MC method, if and only if the Happy set does not have an All-Or-Nothing shape (Definition 3.8), DS-Radial-Shape (Definition 3.14) or ES-radial-shape (Definition 3.23), respectively. Performance indicators are defined, in (4.36) and (4.37), to measure the Robustness optimisation efficiency of the RP methods. Statistical inference on these performance indicators enables the identification of RP methods that perform significantly above average. The conclusions about the empirical results are discussed in Section 6.4.

### Applicability

The mathematical properties of the RP methods are relevant to identify the appropriate Robustness estimation and computation methods, when given an RP problem. Table 6.1 gives an overview of RP problem characteristics, regarding the information about uncontrollable factors $\boldsymbol{v}$ and the structure of the Happy set $\mathbb{H}(x)$.

**Table 6.1**: Applicability of RP methods given RP problem

| Methods | sufficient conditions and requirements on $\boldsymbol{v}$ | sufficient conditions on $\mathbb{H}(x)$ |
|---|---|---|
| MC | Samples of $\boldsymbol{v}$. | Any Happy set based on the RP framework |
| SMC | Samples of $\boldsymbol{v}$. | Any Happy set based on the RP framework |
| N-1MC | CDF of one element of $\boldsymbol{v}$ and samples for the other elements. | A Dim-1-Convex Happy set |
| ES | All elements of $\boldsymbol{v}$ are independent (two-sided) Exponentially distributed | ES-Ray-Convex Happy set (Ray-Convex Happy Happy set) |
| DS | The random vector $\boldsymbol{v}$ has a spherical symmetric distribution | Ray-Convex Happy set |
| Diamond | $R(x)$ estimation: All elements of $\boldsymbol{v}$ are independent (two-sided) Exponentially distributed, or the first two moments of each element of $\boldsymbol{v}$ are given<br>$R(x)$ optimisation: no conditions | Polyhedral Happy set |
| Ball | $R(x)$ estimation: Spherical symmetric distribution with either Normally distributed elements or $t$-distributed elements. Alternatively, the first two moments of each element of $\boldsymbol{v}$ are sufficient<br>$R(x)$ optimisation: no conditions | Polyhedral Happy set |
| Cube | $R(x)$ estimation: All elements of $\boldsymbol{v}$ are independent and either a CDF for each element of $\boldsymbol{v}$ is given or the first two moments of each element of $\boldsymbol{v}$ are given<br>$R(x)$ optimisation: no conditions | Polyhedral Happy set |
| Compression | $\exists_{s\in\mathbb{S}} : u_s(x,\boldsymbol{v}) = w_s(x,\boldsymbol{a})$ where the PDF and CDF of random vector $\boldsymbol{a}$ is known and vector $a$ has A < N independent elements. (Example: Normal i.i.d. $\boldsymbol{v}_n$, $n = 1,..,$N ) | $\exists_{s\in\mathbb{S}} : u_s(x,v) = w_s(x,a)$ where vector $a$ has A < N elements. (Example: Polyhedral Happy set) |
| Decomposition | elements of $\boldsymbol{v}$ are independent | Any Happy set for which the partitions $\mathbb{G}_g(x) \subseteq \mathbb{S}$ for $g = 1,..,G$ exists, according Definition 4.1 |
| WS | $E(\boldsymbol{v})$ is given | Polyhedral Happy set |

## 6.4 Case study results

The fourth research question is how to compare the performance of RP methods. Robustness Programming operates on two levels, i.e. Robustness estimation and Robustness optimisation which is the optimisation of a Robustness estimate function. The performance comparison is based on the following:

- The performance of Robustness estimation methods is assessed by comparing the number of samples that is required to reach a predefined standard error level. The best performing Robustness estimation method is the method that requires the least number of samples

- The performance of Robustness optimisation methods is assessed with performance indicators $\pi$ and $\Pi$ and are respectively defined in (4.42) and (4.43). These performance indicators are based on identifying the RP methods that result in the highest Robustness value found after 20 optimisation iterations. These performance indicators are based on random optimisation starting points. Statistical inference is done to make robust conclusions about the performance indicators $\pi$ and $\Pi$

It is concluded that the computation time (in sec.) and number of floating-point operations (in FLOPS) are not suitable for an objective performance comparison, because the computation time and arithmetic efficiency depends on software quality, computer memory and processor speed. From a practical point of view, however, the computation time or FLOPS are relevant and are given with the remark that they illustrate the performance of the current software implementation and computer configuration, rather than the performance of the RP methods.

Two case studies were carried out for Unilever R&D, one dealing with Robust mixture designs and one dealing with Robust material cost planning. Additionally, a case study is derived from the Robust mixture design case, to assess the performance of the ES method. One case study was done in an Environmental Economics research program called STACO at Wageningen University and deals with the Robustness of coalitions of world regions for jointly reducing $CO_2$ emissions. From these case studies the following can be concluded:

### RP and the STACO case

The STACO case illustrates that the DS method is more efficient per sample than the MC method, for most of the assessed scenarios. However, in the so-called 120% scenario, the Happy set is close to being DS-radially shaped and consequently the efficiency difference between the DS and MC method disappears. The Happy set is not polyhedral and ray intersection distances are determined numerically, which requires a substantial number of FLOPS. From a computational point of view, the measurements support the conclusion that the MC implementation is more efficient than the DS implementation.

### RP and the Unilever case

The Happy sets in the Unilever cases are polyhedral. The empirical results for the Unilever cases support the following conclusions

- The Warm Start method is efficient in combination with the MC, N-1MC, DS and ES method
- Decomposition and Compression are efficient for all cases. However, Decomposition and Compression require additional computations and increase the computation time
- The efficiency of the SMC, N-1MC, DS and ES methods, is above average for at least one of the investigated Unilever cases
- For all Unilever cases, the MC, Ball, Diamond and Cube method are not efficient

## 6.5 An Agenda for Future Research

In the following, some areas are given with respect to opportunities for future research.

### Robustness Programming Framework

- The RP framework is based on $u(x, v)$ being continuous in $v$. Additionally, $u(x, v)$ should also be continuous in $x$, when optimising Robustness with standard NLP methods. Generalising the framework by dropping these continuity conditions has the following relevance.

  For the situation that the controllable factors $x$ can only have discrete values, it is relevant to investigate how Robustness Programming problems can be solved with Combinatorial Optimisation methods.

  The continuity of $u(x, v)$ with respect to $v$ is only relevant for the DS, ES and N-1MC method (if $\boldsymbol{v}$ has a continuous distribution) and is less relevant if $\boldsymbol{v}$ follows a discrete probability distribution.

### Robustness Programming Estimation methods

- The ideas in this study are mainly based on studying the situation that $\boldsymbol{v}$ has a continuous distribution and there is less attention for discrete probability distributions. In principle the N-1 MC method can handle discrete probability distributions. In that situation the N-1MC estimate function is a discontinuous step function of $x$, with problems similar to the problems of the MC method. At this stage the SMC method is most likely the best RP method in the situation that $\boldsymbol{v}$ has a discrete probability distribution. Also the SMC method can have optimisation difficulties, because there is the possibility that at least two samples of $\boldsymbol{v}$ are identical and consequently there exist non standard MC discontinuity points, making the SMC estimate function discontinuous. It is interesting to search for RP alternatives for this situation.

- The smoothing approach in the development of the SMC method gave good results. Possibly this idea is more generally applicable in the context of optimising step functions and other global optimisation problems.

- It is interesting to investigate the possibility to further generalise the DS method, that is based on spherical symmetric distributions, to distributions where the iso-density contours are described by p-norms: let $f$ and $g$ be a PDF such that $f(v) = g(|v|_p)$. It is interesting to investigate whether an unbiased $R(x)$ estimate can be determined as follows:

    1. generate samples $v^{[m]}$, $1, ..,$M
    2. project the samples to $s^{[m]} = \frac{v^{[m]}}{\left\|v^{[m]}\right\|_p}$
    3. rays are defined by half lines through the origin and samples $s^{[m]}$, $1, ..,$M
    4. use the length of the rays in the Happy set, in combination with PDF $g$ to define an unbiased Robustness estimate

- For the DS and ES method, it is essential to have a method to determine the ray intersection distances between $E(\boldsymbol{v})$ and the boundary of $\mathbb{H}(x)$, given a uniform sample on the unit sphere or simplex respectively. If the Happy set is non-polyhedral, it can be a considerable numerical effort to compute such a distance. This negatively affects the computation speed. Additionally, the DS method for example, involves computing a Chi-square CDF value which (depending on the quality of software) can have a significant negative effect on the computation speed. From a sample-size perspective, the MC method is in general less efficient than the DS or ES method. However, from a computation-speed perspective, it is likely that the MC implementation can compute more samples per second than the DS or ES implementation. Consequently, the MC estimate standard error can be smaller than the DS or ES estimate standard error after a fixed number of seconds. For the same reason the SMC method can be more efficient than the DS or ES method. Therefore it is interesting to develop efficient algorithms for determining the ray intersection distances and efficient algorithms for computing the final DS and ES estimate value.

- A special situation for RP is estimating and optimising a Robustness that is close to 1. In such case the MC method is possibly not efficient, because a large number of samples are required to get an accurate estimate. On the other hand, the DS and ES method are possibly the most efficient methods for estimating such probabilities, because an unbiased Robustness estimate can be computed for any ray intersection distance (no matter how many multiples of $\sigma$ are between $E(\boldsymbol{v})$ and the Happy set boundary). It is also interesting to investigate whether the Ball, Diamond and Cube bounding methods have more success, in the context of RP optimisation problems with Robustness values close to 1, than they have shown in the context of the Unilever cases.

- The ideas of Importance Sampling (Law and Kelton, 2000) and Szántai's inclusion-exclusion approach (Gassmann et al., 2002) are interesting approaches and should be exploited further in the context of Robustness Programming. Possibly, combining the bounding methods with Importance Sampling can yield interesting results (also for estimating probabilities close to 1)

### Robustness Computation methods

- The effectiveness of Decomposition is shown for bounding methods and the conditions that lead to strictly sharper bounds are defined. For the estimation methods, the efficiency improvement is only shown empirically for the Unilever cases. It is interesting to find a theoretical proof, showing the efficiency improvement due to Decomposition for estimation methods.

- The Warm Start method is efficient for all estimation methods, except for the SMC method. Furthermore, the applicability of the Warm Start methods is limited to the same class of problems as the bounding methods, since the Warm Start method is based on bounding methods. An interesting alternative is to base the Warm Start method on the SMC method as follows: Let $\hat{R}(x)$ be any of the Robustness estimates for $R(x)$ based on sampling, i.e. an estimate based on the MC, SMC, N-1MC, DS or ES method and redefine the WS estimate function as

$$\overset{\text{ws}}{R}(x) = \begin{cases} \hat{R}(x) & \text{if } \hat{R}(x) > 0 \\ \overset{\text{smc}}{R}(x) & \text{if } \hat{R}(x) = 0 \end{cases}$$

### Case Studies

- Two Unilever cases and the STACO case are investigated during this research. In order to verify the applicability of RP methods on a wider range of cases and to make future research comparable, it is advisable to set up a repository and compile a larger set of reference cases.

- All Robustness optimisations in the case studies are done with the FMINCON solver of Matlab. It is interesting to study Robustness optimisation using other software and optimisation methods and use the comparison methodology of Section 4.6 to determine the best performing optimisation approach.

- An interesting RP case is based on the determination of Economic Capital (EC) in Credit Risk Modelling (Bluhm et al., 2003). It is based on computing $EC_\alpha = VaR_{\boldsymbol{L}}(\alpha) - E(\boldsymbol{L})$, where random variable $\boldsymbol{L}$ is the random loss of a credit portfolio (called loss distribution), $E(\boldsymbol{L})$ is the expected loss of a credit portfolio and $VaR_{\boldsymbol{L}}(\alpha)$ is the Value at Risk with probability $1-\alpha$. A bank can have a customers portfolio of N categories, where $x_n$ for $n = 1, ..,$N is the number of customers in each category. It is possible that a fraction of the portfolio is not able to repay the bank in time and in that situation a loss occurs. A priori, such loss is typically modelled as $\boldsymbol{L}(x)$. For high ranking banks, the $\alpha$-percentile is close to 1. Consequently, MC type of approaches to estimate $VaR_{\boldsymbol{L}(x)}(\alpha)$, require a large number of samples in order to be sufficiently accurate. It is interesting to investigate DS type of methods to estimate loss distribution percentiles more efficiently. Furthermore, the good optimisation characteristics of the DS method can make it possible to develop methods for portfolio optimisation.

# Bibliography

Anderson, T. W. (1984). *An introduction to multivariate statistical analysis (2nd edition).* New York: John Wiley & Sons.

Azaron, A., H. Katagiri, K. Kato, and M. Sakawa (2007, April). A multi-objective discrete reliability optimization problem for dissimilar-unit standby systems. *OR Spectrum 29*(2), 235–257.

Bazaraa, M. S., H. D. Sherali, and C. M. Shetty (1993). *Nonlinear Programming; Theory and Algorithms (2nd edition).* New York: John Wiley & Sons, INc.

Ben-Tal, A. and A. Nemirovski (2002). Robust optimization − methodology and applications. *Mathematical Programming series B 92*, 453–480.

Bernoulli, D. (1738). Specimen theoriae novae de mensura sortis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae.* Translated into English by L. Sommers, 1954, "Exposition of a New Theory on the Measurement of Risk", Econometrica 22, 23−36.

Birge, J. R. and F. Louveaux (1997). *Introduction to Stochastic Programming.* New York: Springer.

Bjerager, P. (1988). Probability integration by directional simulation. *ASCE Journal of Engineering Mechanics 8*(114), 1285–1302.

Bluhm, C., L. Overbeck, and C. Wagner (2003). *An Introduction to Credit Risk Modeling.* Boca Raton, Florida 33431: Chapman & Hall/CRC.

Copeland, T. and V. Antikarov (2001). *Real Options: A Practitioners Guide.* New York: TEXERE.

Dantzig, G. B. (1951). Maximization of a linear function of variables subject to linear inequalities. In Koopmans (Ed.), *Activity Analysis of Production and allocations*, New York. John Wiley & Sons.

Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science* (1), 197–206.

Deák, I. (1986). Computing probabilities of rectangles in case of multinormal distribution. *Journal of Statistical Computation and Simulation* (26), 101–114.

Deák, I. (2000). Subroutines for computing normal probabilities of sets – computer experiences. *Annals of Operations Research 100*(1-4), 103–122.

Deák, I. (2003). Probabilities of simple $n$-dimensional sets for the normal distribution. *IIE Transactions 35*(3), 285–293.

Dellink, R., M. Finus, and N. J. Olieman (2007). The stability likelihood of an international climate agreement. *Environmental and Resource Economics* (Forthcoming).

Devroye, L. (1986). *Non-Uniform Random Variate Generation.* New York: Springer-Verlag.

Du, X. and W. Chen (2000). Towards a better understanding of modeling feasibility robustness in engineering design. *ASME Journal of Mechanical Design 122*(4), 385–394.

Ermoliev, Y. and R. J.-B. Wets (1988). *Numerical Techniques for Stochastic Optimization.* Berlin: Springer-Verlag.

Eyckmans, J. and M. Finus (2003). Coalition formation in a global warming game: How the design of protocols affects the success of environmental treaty-making. Technical Report 56, CLIMNEG 2.

Finus, M., E. C. van Ierland, and R. B. Dellink (2003). Stability of climate coalitions in a cartel formation game. Technical Report 61.2003, Fondazione Eni Enrico Mattei.

Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications (2nd edition).* New York: John Wiley & Sons, Inc.

Franchin, P., O. Ditlevsen, and A. Kiureghian (2002). Model correction factor method for reliability problems involving integrals of non-gaussian random fields. *Probabilistic engineering mechanics* (20), 109–122.

Gassmann, H. I., I. Deák, and T. Szántai (2002). Computing multivariate normal probabilities: A new look. *Journal of Computational and Graphical Statistics 11*(4), 920–949.

Genz, A. (1993). Comparison of methods for the computation of multivariate normal probabilities. *Computing Science and Statistics* (25), 400–405.

Ghahramani, S. (2000). *Fundamentals of Probability Theory.* Upper Saddle River, New Jersey 07458: Prentice Hall.

Goldfarb, D. and G. Iyengar (2003). Robust portfolio selection problems. *Mathematics of Operations research 25*(1), 1–38.

Golub, G. H. and C. F. Van Loan (1989). *Matrix Computations (2nd edition).* Baltimore, Maryland 21211: The Johns Hopkins University Press.

Grimmett, G. and D. Stirzaker (2001). *Probability and Random Processes (3rd edition).* New York: Oxford University Press Inc.

Gurkan, G., Y. Ozge, and S. M. Robinson (1999). Sample path solution of stochastic variational inequalities. *Mathematical Programming 84*, 313–333.

Hendrix, E. M. T. (1998). *Global Optimization at Work.* Ph. D. thesis, Wageningen University.

Hendrix, E. M. T., C. J. Mecking, and T. H. B. Hendriks (1996). Finding robust solutions for product design problems. *European Journal of Operational Research* (92), 28–36.

Hendrix, E. M. T. and N. J. Olieman (2008). The smoothed monte carlo method in robustness optmimisation. *Submitted to 'Optimization Methods and Software'*.

Huele, A. F. (1998). *Statistical Robust Design.* Ph. D. thesis, Universiteit van Amsterdam.

Jacod, J. and P. Protter (2004). *Probability Essentials (2nd edition).* Berlin: Springer-Verlag.

Johnathan, M. (2002). *Real Options Analysis.* New Jersey: John Wiley & Sons, Inc.

Joshi, M. S. (2003). *The concepts and practice of mathematical finance.* Cambridge: Cambridge University Press.

Kall, P. and J. Mayer (1996). Slp-ior: An interactive model management system for stochastic linear programs. *Mathematical Programming* (75), 221–240.

Kelley, C. T. (1999). *Iterative Methods for Optimization.* Philadelphia: SIAM.

Kleijnen, J. P. C., B. Bettonvil, and J. F. Persson (2003, October). Robust solutions for supply chain management: simulation, optimizations and risk analysis. Technical Report robustness_KLICT_1.doc, Center for Economic Research (CentER), Tilbur University.

Kleywegt, A. J., A. Shapiro, and T. Homem de Mello (2001). The sample average approximation method for stochastic discrete optimization. *SIAMJournal of Optimization 12*, 479–502.

Law, A. M. and W. D. Kelton (2000). *Simulation Modeling and Analysis (3rd edition).* Singapore: MacGraw-Hill.

Mak, W. K., D. Morton, and K. Wood (1999). Monte carlo bounding techniques for determining solution quality. *Operations Research Letters 24*(1-2), 47–56.

Markowitz, H. M. (1952). Portfolio selection. *Jounal of Finance 7*, 77–91.

Markowitz, H. M. (1999). The early history of portfolio theory: 1600-1960. *Financial Analysts Journal 55*(4), 5–16.

Marsden, J. E. and A. J. Tromba (1996). *Vector Calculus (4th edition).* New York: W.H. Freeman and Company.

Matousek, J. and J. Nesetril (1998). *Invitation to Discrete Mathematics.* New York: Oxford University Press.

Narayan, C. G. (1996). *Multivariate Statistical Analysis.* 270 Madison Avenue, New York, New York 10016: Marcel Dekker, INC.

Nie, J. and B. R. Ellingwood (2000). Directional methods for structural reliability analysis. *Structural Safety 22*(3), 233–249.

Olieman, N. J. and E. M. T. Hendrix (2005). Global optimisation challenges in robustness programming bounding methods. In *Proceedings of the International Workshop on Global Optimisation*, San José, Almería, Spain.

Olieman, N. J. and E. M. T. Hendrix (2006). Stability likelihood of coalitions in a two-stage cartelgame: An estimation method. *European Journal of Operational Research 174*(1), 333–348.

Olieman, N. J. and B. Van Putten (2006). Estimation method of multivariate exponential probabilities based on a simplex coordinates transform. *Stochastic Programming E-Print Series* (http://www.speps.org).

Osborne, M. J. and A. Rubinstein (1994). *A Course in Game Theory.* Cambridge, Masschusetts / London, England: MIT Press.

Ott, R. L. and M. Longnecker (2001). *An Introduction to Statistical Methods and Data Analysis (5th edition).* Pacific Grove, CA 93950 USA: Duxburry.

Otto, K. N. and E. K. Antonsson (1993). Extensions to the taguchi method of product design. *ASME Journal of Mechanical Design 115*(1), 5–13.

Parkinson, A., C. Sorensen, and N. Pourhassan (1993). A general approach for robust optimal design. *Transactions of the ASME 115*, 74–80.

Prékopa, A. (2001). On the concavity of multivariate probability distribution functions. *Operations research letters* (29), 1–4.

Randolph, J. F. (1968). *Basic Real and Abstract Analysis.* New York: Academic Press.

Rasmusen, E. (1994). *Games and Information (2nd edition).* 238 Main Street, Cambridge, Massachusetts 02142, USA: Blackwell Publishers.

Rausand, M. and A. Høyland (2004). *System Reliability Theory: Models, Statistical Methods, and Applications (2nd edition).* Hoboken, New Jersey: John-Wiley & Sons. Inc.

Rice, J. A. (1995). *Mathematical statistics and data analysis (2nd edition).* Belmont, California 94002: Wadsworth Publishing Company.

Robert, C. P. and G. Casella (1999). *Monte Carlo Statistical Methods.* New York: Springer-Verlag.

Rubinstein, M. (2002). Markowitz's "portfolio selection": A fifty-year retrospective. *Jounal of Finance 57*(3), 1041–1045.

Sáiz, M. E., E. M. T. Hendrix, and N. J. Olieman (2006). On the computation of stability in multiple coalition formation games. *Computational Economics 25*(3), 251–275.

Schittkowski, K. (1985). Nlqpl: A fortran-subroutine solving constrained nonlinear programming problems. *Annals of Operations Research 5*, 485–500.

Schittkowski, K. (1994). Numerical comparison of nonlinear programming algorithms for structural optimization. *Structural and Multidisciplinary Optimization 7*(1), 1–19.

Schittkowski, K. (2005). Optimization in industrial engineering: Sqp-methods and applications. Technical report, Department of Computer Science, University of Bayreuth, D - 95440 Bayreuth, Germany.

Shakespeare, W. (1914). *The Complete works of William Shakespeare*. London: Oxford University Press.

Silver, E. A., D. F. Pyke, and R. Peterson (1998). *Inventory Management and Production Planning and Scheduling (3rd edition)*. New York: John Wiley & Sons.

Somerville, P. N. (1998). Numerical computation of multivariate normal and multivariate-t probabilities over convex regions. *Journal of Computational and Graphical Statistics Volume 7*(Number 4), 529–544.

Szántai, T. (2000). Improved bounds and simulation procedures on the value of the multivariate normal probability distribution function. *Annals of Operations Research 100*(1-4), 85–101.

Taguchi, G. (1986). *Introduction to Quality Engineering: Designing Quality into Products and Processes*. Tokyo: Asian Productivity Organization.

Terje, A. and J. Uwe (1999). *Stochastic Models in Reliability*. New York: Springer-Verlag.

Tijms, H. C. (2002). *Operationele Analyse (2nd edition)*. Utrecht, the Netherlands: Epsilon Uitgaven.

Van Loan, C. F. (2000). *Introduction to Scientific Computing (2nd edition)*. Upper Saddle River, NJ 07458: Prentice-Hall.

Venkataraman, P. (2002). *Applied Optimization with MATLAB® Programming*. New York: John Wiley & Sons.

Verbeek, M. (2004). *A guide to modern econometrics (2nd edition)*. West Sussex, England: John Wiley & Sons, Ltd.

Weisstein, E. W. (1998). *"Normal Distribution." From MathWorld–A Wolfram Web Resource*. http://mathworld.wolfram.com/SmoothFunction.html: Wolfram Research, Inc.

Winston, W. L. (1993). *Operations Research: applications and algorithms (3rd edition)*. Belmonte, California 94002, USA: Wadsworth Publishing Company.

Yi, S. (1997). Stable coalition structures with externalities. *Games and Economic Behavior* (20), 201–237.

# Appendix

## A.1 Nomenclature

| | |
|---|---|
| $x$ | design vector consisting of elements $x_i$, with $i = 1, ..,$I |
| $X$ | design matrix reflecting K alternative designs, consisting of elements $X_{i,k}$, with $k = 1, ..,$K and $i = 1, ..,$I |
| $\boldsymbol{v}$ | stochastic vector consisting of independent elements $\boldsymbol{v}_n$, with $n = 1, .., \mathrm{N}$ and $E(\boldsymbol{v_n}) = 0$, $VAR(\boldsymbol{v_n}) = 1$ and $COV[\boldsymbol{v_n}, \boldsymbol{v_p}] = 0$, for $n \neq p$, unless specified otherwise |
| $v$ | realisation of the stochastic vector $\boldsymbol{v}$ |
| $\mathbb{V}$ | sample space of $\boldsymbol{v}$. Hence $v \in \mathbb{V}$ |
| $\mathbb{X}$ | set of feasible designs, with $\mathbb{X} \subseteq \mathbb{R}^{\mathrm{I}}$ |
| $\boldsymbol{\chi}^2(\mathrm{N})$ | Chi-square distributed variate with N degrees of freedom.<br>(relation: $\boldsymbol{\chi}^2(\mathrm{N}) \sim \sum_{i=1}^{\mathrm{N}} (\boldsymbol{\chi}_i)^2$, with $\boldsymbol{\chi_i} \sim N(0, 1)$ |
| $\boldsymbol{F}(\delta_1, \delta_2)$ | F-distributed variate with $\delta_1$ and $\delta_2$ degrees of freedom.<br>(relation: $\boldsymbol{F}(\delta_1, \delta_2) \sim \frac{\boldsymbol{\chi}^2(\delta_1)/\delta_1}{\boldsymbol{\chi}^2(\delta_2)/\delta_2})$ |
| $\mathbb{H}(x)$ | the Happy set as function of $x$ where $\mathbb{H}(x) \subseteq \mathbb{R}^{\mathrm{N}}$. |
| $\partial\mathbb{H}(x)$ | boundary of the Happy set. |
| i.i.d. | independent identically distributed |
| w.r.t | with respect to |
| a.s. | almost surely |
| a.e. | almost everywhere |

## A.2 Probability Space

Based on Probability Theory definitions as presented in Ghahramani (2000); Grimmett and Stirzaker (2001); Jacod and Protter (2004).

1. The abstract probability space is defined as $(\Omega, \mathcal{F}, \mathrm{Pr})$, with *sample space* $\Omega$; the smallest $\sigma$-field $\mathcal{F}$ of all subsets of $\Omega$; a probability measure $\mathrm{Pr} : \mathcal{F} \to [0, 1]$. All subsets $\mathbb{A} \in \mathcal{F}$ of $\Omega$ are called *events* and a particular $\omega \in \Omega$ is called an *realisation* of the random experiment $\boldsymbol{\omega}$. The probability of event $\mathbb{A} \in \mathcal{F}$ is denoted with $\mathrm{Pr}(\mathbb{A})$.

2. By definition $\mathrm{Pr}\,(\emptyset) = 0$, $\mathrm{Pr}\,(\Omega) = 1$ and $0 \leq \mathrm{Pr}\left(\bigcup_i \mathbb{A}_i\right) = \sum_i \mathrm{Pr}\,(\mathbb{A}_i) \leq 1$ for any disjoint sets $\mathbb{A}_1, \mathbb{A}_2, ...\mathbb{A}_i$ in $\mathcal{F}$ and $\mathrm{Pr}\,(\mathbb{B}_1) \leq \mathrm{Pr}\,(\mathbb{B}_2) \leq \mathrm{Pr}\,(\mathbb{B}_3) \leq ... \leq \mathrm{Pr}\,(\mathbb{B}_i)$ for an increasing sequence of sets $\mathbb{B}_1 \subseteq \mathbb{B}_2 \subseteq \mathbb{B}_3 \subseteq ... \subseteq \mathbb{B}_i$ in $\mathcal{F}$.

3. The N-dimensional random vector $v(\boldsymbol{\omega})$ is a Borel measurable mapping[2] $v : \Omega \longrightarrow \mathbb{R}^{\mathrm{N}}$. The probability measure induced by the random vector $v(\boldsymbol{\omega})$ is defined as

[2] Mapping $v$ is Borel measurable if $\{\omega \in \Omega | v(\omega) \in \mathbb{S}\} \in \mathcal{B}(\Omega)$ for all $\mathbb{S} \in \mathcal{B}(\mathbb{R}^{\mathrm{N}})$ where $\mathcal{B}(\cdot)$ defines the class of all Borel sets.

$\Pr(\{\boldsymbol{\omega} \in \Omega | v(\boldsymbol{\omega}) \in \mathbb{S}\})$, with $\mathbb{S} \in \mathcal{B}(\mathbb{R}^{\mathrm{N}})$[3]. The random vector $v(\boldsymbol{\omega})$ has *possible outcomes* $v(\omega) \in v(\Omega) = \mathbb{V} \subseteq \mathbb{R}^{\mathrm{N}}$, with $\omega \in \Omega$.

4. The notation $\boldsymbol{v}$ is a shorthand notation for $v(\boldsymbol{\omega})$. The random vector $\boldsymbol{v}$ is defined on probability space $(\mathbb{V}, \mathcal{V}, \Pr_{\boldsymbol{v}})$ with $\mathcal{V} = \mathcal{B}(\mathbb{V})$ and $\Pr_{\boldsymbol{v}}(\mathbb{S}) = \Pr(\{\boldsymbol{\omega} \in \Omega | v(\boldsymbol{\omega}) \in \mathbb{S}\})$. The notation $\Pr\{\boldsymbol{v} \in \mathbb{S}\} = \Pr_{\boldsymbol{v}}(\mathbb{S})$ is also used.

## A.3 Accurate Robustness estimate

Accurate estimate $\tilde{R}(x) = \overset{\mathrm{mc}}{R}(x)$ is defined in two versions. In version 1, an upper bound on the standard error determines the number of samples to use. In version 2, the number of samples is fixed resulting in a standard error.

1. In the first version, $\hat{se}\left(\tilde{\boldsymbol{R}}(x)\right) < \overline{\sigma}$ , where $\overline{\sigma}$ is the accuracy level. At most M= $1,000,001$ samples are needed to reach an high accuracy like $\overline{\sigma} = 0.0005$. For the estimated standard error holds that

$$\hat{se}\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right) = \sqrt{\frac{1}{\text{M-1}}\left(\overset{\mathrm{mc}}{R}(x) - \overset{\mathrm{mc}}{R}(x)^2\right)} \leq \sqrt{\frac{1}{\text{M-1}}\left(\max_{0\leq p\leq 1}(p-p^2)\right)} = \sqrt{\frac{0.25}{\text{M-1}}}.$$

M= $1,000,001$ gives that $\sqrt{\frac{0.25}{\text{M-1}}} = 0.0005$. On the other hand, the situation can occur that $\tilde{R}(x) = 0$ or $\tilde{R}(x) = 1$ based on M samples. In such case $\hat{se}\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right) = 0$. Such standard error estimate may lead to misinterpretation, especially if M is small. For example let $R(x) = 0.8$ and M=4, then it is possible that all 4 samples are in the happy set. Consequently, $\tilde{R}(x) = 1$ and $\hat{se}\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right) = 0$, based on these 4 samples. A practical solution is the following. One can claim that, a priori verifying the next sample (M+1), we know that $0 \leq \hat{se}\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right) \leq \sqrt{\frac{1}{\text{M}}\left(\frac{1}{\text{M+1}} - \frac{1}{(\text{M+1})^2}\right)} = \frac{1}{\text{M+1}}$. Interestingly, the interval becomes smaller as M increases.

In general we define $\tilde{\boldsymbol{R}}(x) = \overset{\mathrm{mc}}{\boldsymbol{R}}(x)$ with M= $\min\left\{\text{M} \in \mathbb{N} \middle| \hat{se}\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right) \leq \overline{\sigma}, \frac{1}{\text{M+1}} \leq \overline{\sigma}\right\}$. Hence, the computation time of $\tilde{R}(x)$ depends on the value of $\tilde{R}(x)$, because the estimate determines how many samples are needed to reach the accuracy.

2. In the second version, the number of samples M is fixed. Consequently,

$$\hat{se}\left(\overset{\mathrm{mc}}{\boldsymbol{R}}(x)\right) = \sqrt{\frac{1}{\text{M-1}}\left(\overset{\mathrm{mc}}{R}(x) - \overset{\mathrm{mc}}{R}(x)^2\right)} \leq \sqrt{\frac{0.25}{\text{M-1}}}$$

[3] $\mathcal{B}(\mathbb{R}^{\mathrm{N}})$ is the class of all Borel subsets of $\mathbb{R}^{\mathrm{N}}$ and is defined as the $\sigma$-field generated by all open and closed subsets of $\mathbb{R}^{\mathrm{N}}$.

## A.4 Advanced SMC estimation method

The SMC method as discussed in 3.4 can be modified, maintaining its continuity properties for the situation that all samples are either inside or outside the Happy set. In Section 3.4 the smoothing term $s(x)$ has been defined. The extension is based on an alternative smoothing term $t(x)$:

$$\overset{\text{smc}}{R}(x) = \overset{\text{mc}}{R}(x) + t(x) \tag{A.1}$$

where $t : \mathbb{X} \longrightarrow [-\frac{1}{2M}, \frac{1}{2M})$ is a modification of the smoothing function $s$ defined as

$$t(x) = \begin{cases} -\frac{1}{2M(d^{[in]}(x)+1)} & if \quad \overset{\text{mc}}{R}(x) = 1 \\ \frac{1}{2M(d^{[out]}(x)+1)} & if \quad \overset{\text{mc}}{R}(x) = 0 \\ s(x) & otherwise \end{cases} \tag{A.2}$$

## A.5 Robustness of Diamond set given two-sided Exponential distribution

**Theorem A.1** *Let $\boldsymbol{v}_n$ be independently two-sided exponentially distributed with PDF:*

$$f(v_n) = \begin{cases} \frac{1}{2}e^{-v_n} & for \quad v_n \geq 0 \\ \frac{1}{2}e^{v_n} & for \quad v_n < 0 \end{cases} \tag{A.3}$$

*The expected value is $E(\boldsymbol{v}_n) = 0$ and variance is $VAR(\boldsymbol{v}_n) = 2$ for $n = 1, .., N$. Then*

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r)\right\} = 1 - e^{-r}\sum_{i=0}^{N-1}\frac{r^i}{i!}$$

*with*

$$\mathbb{B}^{[1]}(r) = \left\{ v \in \mathbb{R}^N \middle| \sum_{n=1}^{N} |v_n| \leq r \right\}$$

**Proof.** Both the Diamond set $\mathbb{B}^{[1]}(r)$ and the multivariate two-sided exponential distribution are point symmetric in the origin, such that

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}\right\} = \frac{1}{2^{\mathrm{N}}}\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r)\right\}$$

where the part of the diamond in the positive orthant is

$$\mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}} = \left\{ v \in \mathbb{R}_+^{\mathrm{N}} \middle| \sum_{n=1}^{\mathrm{N}} v_n \leq r \right\}$$

The approach is first to find an analytic expression for the probability mass of the Diamond in the positive orthant and then correct with the factor $2^{\mathrm{N}}$ to find the probability mass of the Diamond. In Section 3.7 the Simplex Coordinates transform

$$T : \mathbb{Q} \longrightarrow \mathbb{R}_+^{\mathrm{N}} \text{ with} \tag{A.4}$$

$$T(\rho, s_1, s_2, ..., s_{\mathrm{N}-1}) = \left(\rho s_1, \rho s_2, ..., \rho s_{\mathrm{N}-1}, \rho(1 - \sum_{i=1}^{\mathrm{N}-1} s_i)\right) \tag{A.5}$$

has been introduced, where the determinant of the Jacobian is $(-1)^{\mathrm{N}} r^{\mathrm{N}-1}$ and $\mathbb{Q} = \left(\mathbb{R}_+, \mathbb{S}^{\mathrm{N}-1}\right)$ and $\mathbb{S}^{\mathrm{N}-1} = \left\{ \boldsymbol{s} \in \mathbb{R}_+^{\mathrm{N}-1} \middle| \sum_{i=1}^{\mathrm{N}-1} s_i \leq 1 \right\}$. Application of the Simplex Coordinates transform to

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}\right\} = \int\limits_{\mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}} \prod_{n=1}^{\mathrm{N}} \frac{1}{2} e^{-v_n} dv = \frac{1}{2^{\mathrm{N}}} \int\limits_{\mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}} e^{-\sum\limits_{n=1}^{\mathrm{N}} v_n} dv$$

gives

$$2^{\mathrm{N}} \Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}\right\} = \int\limits_{\mathbb{S}^{\mathrm{N}-1}} \int\limits_0^{\infty} I^{[SC]}(\rho, s) e^{-\rho} \rho^{\mathrm{N}-1} d\rho ds \tag{A.6}$$

where factor $e^{-\rho}$ is obtained via

$$2^{\mathrm{N}} f\left(T(\rho, \boldsymbol{s})\right) = \exp\left(-\sum_{i=1}^{\mathrm{N}-1} \rho s_i\right) \exp\left(-\rho(1 - \sum_{i=1}^{\mathrm{N}-1} s_i)\right) = \exp(-\rho) \tag{A.7}$$

for $\rho \in \mathbb{R}_+$ and $s \in \mathbb{S}^{\mathrm{N}-1}$ and where $I^{[SC]} : \mathbb{Q} \longrightarrow \{0, 1\}$ is the indicator function of $\mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}$, which expressed in Simplex Coordinates (SC) reads

$$I(\rho, s) = \begin{cases} 1 & \text{if } T(\rho, s) \in \mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}} \\ 0 & \text{elsewhere} \end{cases} . \tag{A.8}$$

By definition $T(\rho, s) \in \mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}$ if and only if $\rho \leq r$ and $s \in \mathbb{S}^{\mathrm{N}-1}$. Consequently

$$2^{\mathrm{N}} \Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r) \bigcap \mathbb{R}_+^{\mathrm{N}}\right\} = \int\limits_{\mathbb{S}^{\mathrm{N}-1}} \int\limits_0^{r} e^{-\rho} \rho^{\mathrm{N}-1} d\rho ds \tag{A.9}$$

$$= \int\limits_{\mathbb{S}^{\mathrm{N}-1}} ds \int\limits_0^{r} e^{-\rho} \rho^{\mathrm{N}-1} d\rho. \tag{A.10}$$

From standard calculus follows that the volume in the positive orthant, bounded by a unit simplex in dimension N−1 is

$$\int\limits_{\mathbb{S}^{\mathrm{N}-1}} ds = \frac{1}{(\mathrm{N}-1)!} \tag{A.11}$$

such that the probability mass of the diamond can be expressed analytically:

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[1]}(r)\right\} = \frac{1}{(\mathrm{N}-1)!}\int\limits_{0}^{r} e^{-\rho}\rho^{\mathrm{N}-1}d\rho \qquad = 1 - e^{-r}\sum_{i=0}^{\mathrm{N}-1}\frac{r^i}{i!} \tag{A.12}$$

■

## A.6 Weakness of origin condition

**Lemma A.1** *Let set $\mathbb{S} \subset \mathbb{R}^N$ be convex and $\boldsymbol{v}$ an $N$-dimensional random vector, with centrally i.i.d elements and a probability density function with the property that $f(v) = f(-v)$ (point symmetric in the origin).*

$$\text{If } E(\boldsymbol{v}) \notin \mathbb{S} \text{ then } \Pr\{\boldsymbol{v} \in \mathbb{S}\} < \frac{1}{2} \tag{A.13}$$

**Proof.** From $E(\boldsymbol{v_i}) = 0$ follows $E(c'\boldsymbol{v}) = 0$, for all $c \in \mathbb{R}^{\mathrm{N}}$. The symmetry property $f(v) = f(-v)$ implies that for a $c \in \mathbb{R}^{\mathrm{N}}$:

$$\begin{aligned} &\Pr\{c'\boldsymbol{v} \geq 0\} = \Pr\{-c'\boldsymbol{v} \geq 0\} = \Pr\{c'\boldsymbol{v} \leq 0\} \\ \Longrightarrow &\Pr\{c'\boldsymbol{v} \geq 0\} = 1 - \Pr\{c'\boldsymbol{v} \geq 0\} \\ \Longrightarrow &\Pr\{c'\boldsymbol{v} \geq 0\} = \frac{1}{2} \end{aligned}$$

In the remainder of this proof, we use the idea of a *separating hyper-plane.* Let $v^*$ be a point closest to the origin $E(\boldsymbol{v})$:

$$v^* = \arg\inf_{v \in \mathbb{S}} v'v \tag{A.14}$$

with $d = \sqrt{v^{*\prime}v^*}$. From the convexity of $\mathbb{S}$ follows that a tangent plane $v^{*\prime}v = r^2$ does not intersect $\mathbb{S}$, if $r^2 < d^2$. In particular this holds for the tangent plane through the point $E(\boldsymbol{v})$, i.e. $v^{*\prime}v = 0$. Consequently the set $\mathbb{S}$ is a *strict* subset of the half space $\mathbb{H}$:

$$\mathbb{S} \subset \mathbb{H} = \left\{v \left| v^{*\prime}v \geq 0\right.\right\}$$

which implies implies that:

$$\Pr\{\boldsymbol{v} \in \mathbb{S}\} < \Pr\{\boldsymbol{v} \in \mathbb{H}\} = \Pr\left\{v^{*\prime}\boldsymbol{v} \geq 0\right\} = \frac{1}{2}$$

■

## A.7 Markov Inequality applied to ball probability

**Theorem A.2** *Let $\boldsymbol{v}_n$ be independent random variables with $E(\boldsymbol{v}_n) = 0$, $VAR(\boldsymbol{v}_n) = 1$ and $n = 1,..,\mathrm{N}$. Then*

$$\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[2]}(\rho)\right\} = \Pr\left\{\sum_{n=1}^{\mathrm{N}}\boldsymbol{v}_n^2 \leq \rho^2\right\} \geq 1 - \frac{\mathrm{N}}{\rho^2} \tag{A.15}$$

**Proof.** From the first two moments of the elements of the random vector follows that $E(\boldsymbol{v}_n^2) = (E(\boldsymbol{v}_n))^2 + VAR(\boldsymbol{v}_n) = 1$ with $n = 1,..,\mathrm{N}$ and consequently $E(\sum\limits_{n=1}^{\mathrm{N}} \boldsymbol{v}_n^2) = \mathrm{N}$. Since $\boldsymbol{v}_n^2$ is non-negative by definition, we can apply Markov's Inequality (which follows from the results in Ghahramani, 2000):

$$\Pr\left\{\sum_{n=1}^{\mathrm{N}} \boldsymbol{v}_n^2 \leq \rho^2\right\} \geq 1 - \frac{E\left(\sum\limits_{n=1}^{\mathrm{N}} \boldsymbol{v}_n^2\right)}{\rho^2} = 1 - \frac{\mathrm{N}}{\rho^2} \tag{A.16}$$

■

## A.8 Effect of dimension N on DS efficiency

Chapter 3 has shown that the accuracy of the DS Robustness estimator tends to the efficiency of the MC estimator if the DS estimator has a high probability on the two outcomes in $\{0, 1\}$. The following example shows a case where the DS method tends to $\{0, 1\}$ outcomes, with increasing dimension. Let

$$\mathbb{H}(x) = \left\{v \in \mathbb{R}^{\mathrm{N}} | -1.5 \leq \sum_{n=1}^{\mathrm{N}} v_n x_n \leq 1.5\right\} \tag{A.17}$$

where, $x_1 = 1$, $x_i = 0$, for $i = 2,..,\mathrm{I} = \mathrm{N}$ and $\boldsymbol{v}_n \sim N(0,1)$ for $n = 1,..,\ \mathrm{N}$. Let $\boldsymbol{s}$ be a uniformly distributed random vector on the unit sphere. Let $r(s)$ be the distance between the origin and the point where the vector through $s$ intersects the bounding plane of the Happy set. From Deák's estimate function (3.69) on page 52 follows that the unbiased Robustness estimate is:

$$\overset{\mathrm{ds}}{r}(s, x) = \Pr\left\{\boldsymbol{\chi}^2(\ \mathrm{N}) \leq r^2(s)\right\} \tag{A.18}$$

where $\boldsymbol{\chi}^2(\ \mathrm{N})$ is a chi-square distributed random variable with N degrees of freedom. A nice feature of the Directional Sampling method, is that estimates typically have values smaller than 1 and bigger than 0, resulting in a smaller standard error for the DS method compared to the MC method. For this example this nice feature is vanishing with increasing dimension N:

- A uniformly distributed random vector $\boldsymbol{s}$ on the unit sphere can be obtained via i.i.d. $\boldsymbol{y}_n \sim N(0,1)$ for $n = 1,..,\ \mathrm{N}$ and normalised to $\boldsymbol{s} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|}$.
- The squared distance to the boundary of Happy set (A.17) for a realisation $s \in \mathbb{R}^{\mathrm{N}}$ of $\boldsymbol{s}$, is defined by the first coordinate of the point on the unit sphere:

$$r^2(s) = \begin{cases} \left(\frac{1.5}{s_1}\right)^2 & \text{for } s_1 \neq 0 \\ \infty & \text{for } s_1 = 0 \end{cases}$$

- For $\mathrm{N} > 1$, the random squared distance $r^2(\boldsymbol{s})$ is distributed as

$$r^2(\boldsymbol{s}) = r^2(\frac{\boldsymbol{y}_1}{\|\boldsymbol{y}\|}) = \frac{1.5^2 \sum_{n=1}^{\mathrm{N}} (\boldsymbol{y}_n)^2}{(\boldsymbol{y}_1)^2} = 1.5^2 \left(1 + \frac{\boldsymbol{\chi}^2(\mathrm{N}-1)}{\boldsymbol{\chi}^2(1)}\right) = 1.5^2 \left(1 + \frac{\mathrm{N}-1}{\boldsymbol{t}^2_{\mathrm{N}-1}}\right) \tag{A.19}$$

  where $\boldsymbol{t}_k$ is the student t-distribution with $k$ degrees of freedom.

- Let $\Psi_{\mathrm{N}}(w) = \Pr\{\boldsymbol{\chi}^2(\mathrm{N}) \leq w\}$ be the CDF of $\boldsymbol{\chi}^2(\mathrm{N})$. The DS estimator can be expressed as $\overset{\mathrm{ds}}{r}(\boldsymbol{s}, x) = \Psi_{\mathrm{N}}(r^2(\boldsymbol{s})) = \Psi_{\mathrm{N}}\left(1.5^2\left(1 + \frac{\mathrm{N}-1}{\boldsymbol{t}^2_{\mathrm{N}-1}}\right)\right)$

- From The Central Limit Theorem (Rice, 1995) and $E(\boldsymbol{\chi}^2(\mathrm{N})) = \mathrm{N}$ follows that for constants $\alpha$ and $\beta$:

$$\lim_{\mathrm{N}\to\infty} \Psi_{\mathrm{N}-1}(\beta + \alpha(\mathrm{N}-1)) = \begin{cases} \frac{1}{2} & for \quad \alpha = 1 \\ 1 & for \quad \alpha > 1 \\ 0 & for \quad \alpha < 1 \end{cases} \tag{A.20}$$

From Equations (A.20) and (A.19) follows that:

- The realisation $s$ corresponding to realisation $t_{\mathrm{N}-1} \in \left\{t_{\mathrm{N}-1} \in \mathbb{R} | t^2_{\mathrm{N}-1} < 1.5^2\right\}$ of random variable $\boldsymbol{t}_{\mathrm{N}-1}$ results in $\lim_{\mathrm{N}\to\infty} \overset{\mathrm{ds}}{r}(s, x) = 1$

- The realisation $s$ corresponding to realisation $t_{\mathrm{N}-1} \in \left\{t_{\mathrm{N}-1} \in \mathbb{R} | t^2_{\mathrm{N}-1} > 1.5^2\right\}$ of random variable $\boldsymbol{t}_{\mathrm{N}-1}$ results in $\lim_{\mathrm{N}\to\infty} \overset{\mathrm{ds}}{r}(s, x) = 0$

Consequently, if $\mathrm{N} \to \infty$ then $\Pr\left\{0 < \overset{\mathrm{ds}}{r}(\boldsymbol{s}, x) < 1\right\} \to 0$. This means that the standard error of the DS estimator approaches the standard error of the MC estimator.

## A.9 Decomposition and effective bounding

The 1-norm, 2-norm and $\infty$-norm, are special and limit cases of the Hölder $p$-norm: $\|v\|_p = \left(\sum_{n=1}^{\mathrm{N}} |v_n|^p\right)^{\frac{1}{p}}$

**Lemma A.2** *Let $a \in \mathbb{R}^{\mathrm{A}}$ and $b \in \mathbb{R}^{\mathrm{B}}$. Then*

$$\begin{aligned} \|(a_1, \ldots, a_{\mathrm{A}}, b_1, \ldots, b_{\mathrm{B}})\|_p &= \left(\sum_{i=1}^{\mathrm{A}} |a_i|^p + \sum_{i=1}^{\mathrm{B}} |b_i|^p\right)^{\frac{1}{p}} \\ &= \left(\left(\sqrt[p]{\sum_{i=1}^{\mathrm{A}} |a_i|^p}\right)^p + \left(\sqrt[p]{\sum_{i=1}^{\mathrm{B}} |b_i|^p}\right)^p\right)^{\frac{1}{p}} \\ &= \left\|\left(\|a\|_p, \|b\|_p\right)^{\intercal}\right\|_p \end{aligned} \tag{A.21}$$

**Remark A.3** *(Minkowski Inequality (Randolph, 1968))*

*Let $a, b \in \mathbb{R}^{\mathrm{N}}$. Then*

$$\|a+b\|_p \leq \|a\|_p + \|b\|_p \tag{A.22}$$

Consider the definitions of $\overline{\mathbb{H}}^{[p]}$ with $p \in \{1, 2, \infty\}$ as in (3.97) which are respectively the smallest diamond, ball and cube with radius $\overline{r}^{[p]}$ that enclose the Happy set $\mathbb{H}$. Consider the idea described in Section 4.3.1 of *decomposing* $\overline{\mathbb{H}}^{[p]}$ into lower dimensional sets $\overline{\mathbb{H}}_g^{[p]} \subseteq \mathbb{R}^{\mathrm{M}_g}$, $g = 1, .., \mathrm{G}$. Theorem 4.1 and Corollary 4.5 explain why and when decomposition is effective.

**Theorem 4.1**

Let $\|\cdot\|$ be the notation for a p-norm $\|\cdot\|_p$. Let $\bigcup\limits_{g=1}^{\mathrm{G}} \mathbb{G}_g(x) = \mathbb{S}$ be the decomposition of the uncertain object properties as given in Definition 4.1. Let $\left(v^{[1]^\intercal}, .., v^{[g]^\intercal}, ..v^{[\mathrm{G}]^\intercal}\right)^\intercal \in \mathbb{R}^{\mathrm{N}}$ be the corresponding decomposition of $v$, where $\mathrm{N}= \sum\limits_{g=1}^{\mathrm{G}} \mathrm{M}_g$, $v \in \mathbb{R}^{\mathrm{N}}$ and $v^{[g]} \in \mathbb{R}^{\mathrm{M}_g}$. Let $P^{[g]}$ be a $\mathrm{N}{\times}\mathrm{M}_g$ projection matrix, such that $v = \sum\limits_{g=1}^{\mathrm{G}} P^{[g]} v^{[g]}$, $v^{[g]} = P^{[g]^\intercal} v$ and $\left\|P^{[g]} v^{[g]}\right\| = \left\|v^{[g]}\right\|$. Let $\mathbb{H} \subset \mathbb{R}^{\mathrm{N}}$ be a set that can be decomposed as $\mathbb{H} = \bigcap\limits_{g=1}^{\mathrm{G}} \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| P^{[g]^\intercal} v \in \mathbb{H}_g \right\}$, with $\mathbb{H}_g \subseteq \mathbb{R}^{\mathrm{M}_g}$. Let $\overline{r} = \max\limits_{v \in \mathbb{H}} \|v\|$ define the radius of the smallest set $\overline{\mathbb{H}} = \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \|v\| \leq \overline{r} \right\}$ enclosing the Happy set, i.e.$\overline{\mathbb{H}} \supseteq \mathbb{H}$. Similarly, let $\overline{r}_g = \max\limits_{v^{[g]} \in \mathbb{H}_g} \left\|v^{[g]}\right\| = \max\limits_{v \in \mathbb{H}} \left\|P^{[g]^\intercal} v\right\|$ define $\overline{\mathbb{H}}_g = \left\{ v^{[g]} \in \mathbb{R}^{\mathrm{M}_g} \middle| \left\|v^{[g]}\right\| \leq \overline{r}^{[g]} \right\} \supseteq \mathbb{H}_g$ for $g = 1, .., \mathrm{G}$. Let $\boldsymbol{v}$ have independent distributed elements and PDF $f(v)$ with $f(v) > 0$ for all $v \in \mathbb{R}^{\mathrm{N}}$. Then

$$\prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}} \left(\overline{\mathbb{H}}_g\right) \leq \Pr_{\boldsymbol{v}} \left(\overline{\mathbb{H}}\right) \tag{A.23}$$

**Proof.**

$$\begin{aligned}\prod_{g=1}^{\mathrm{G}} \Pr_{\boldsymbol{v}^{[g]}} \left(\overline{\mathbb{H}}_g\right) &= \Pr_{\boldsymbol{v}} \left( \bigcap_{g=1}^{\mathrm{G}} \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| P^{[g]^\intercal} v \in \overline{\mathbb{H}}_g \right\} \right) \\ &= \Pr_{\boldsymbol{v}} \left( \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \forall_{g=1,..,\mathrm{G}} : P^{[g]^\intercal} v \in \overline{\mathbb{H}}_g \right\} \right) \end{aligned} \tag{A.24}$$

Theorem 4.1 holds if

$$\left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \forall_{g=1,..,\mathrm{G}} : P^{[g]^\intercal} v \in \overline{\mathbb{H}}_g \right\} \subseteq \overline{\mathbb{H}} \tag{A.25}$$

By definition

$$\left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \forall_{g=1,..,\mathrm{G}} : P^{[g]^\intercal} v \in \overline{\mathbb{H}}_g \right\} = \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \forall_{g=1,..,\mathrm{G}} : \left\|P^{[g]^\intercal} v\right\| \leq \overline{r}_g \right\} \tag{A.26}$$

Thus, it remains to be shown that

$$\left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \forall_{g=1,..,\mathrm{G}} : \left\| P^{[g]^{\mathsf{T}}} v \right\| \leq \overline{r}_g \right\} \subseteq \left\{ v \in \mathbb{R}^{\mathrm{N}} \,|\, \|v\| \leq \overline{r} \right\} \tag{A.27}$$

From Lemma A.2 follows that

$$\begin{aligned} \|v\| &= \left\| \left( \left\|v^{[1]}\right\|, \cdots, \left\|v^{[g]}\right\|, \cdots, \left\|v^{[\mathrm{G}]}\right\| \right) \right\| \\ &= \left\| \left( \left\|P^{[1]^{\mathsf{T}}} v\right\|, \cdots, \left\|P^{[g]^{\mathsf{T}}} v\right\|, \cdots, \left\|P^{[\mathrm{G}]^{\mathsf{T}}} v\right\| \right) \right\| \end{aligned} \tag{A.28}$$

from which follows that

$$\begin{aligned} \overline{r} &= \max_{v \in \mathbb{H}} \|v\| \\ &= \left\| \left( \max_{v \in \mathbb{H}} \left\|P^{[1]^{\mathsf{T}}} v\right\|, \cdots, \max_{v \in \mathbb{H}} \left\|P^{[g]^{\mathsf{T}}} v\right\|, \cdots, \max_{v \in \mathbb{H}} \left\|P^{[\mathrm{G}]^{\mathsf{T}}T} v\right\| \right) \right\| \\ &= \left\| \left( \overline{r}_1, .., \overline{r}_g, .., \overline{r}_{\mathrm{G}} \right) \right\| \end{aligned} \tag{A.29}$$

The result (A.27) follows from the observation that for all $v$ for which $\forall_{g=1,..,\mathrm{G}} : \left\| P^{[g]^{\mathsf{T}}} v \right\| \leq \overline{r}_g$, holds $\|v\| \leq \overline{r}$, i.e. satisfying the condition in the left-hand side of A.27 implies satisfying the condition in the right-hand side of A.27. ■

**Corollary 4.5** If there exists a $g \in \{1, .., \mathrm{G}\}$ with $\overline{r}_g < \overline{r}$ then

$$\prod_{g=1}^{\mathrm{G}} \mathrm{Pr}_{\boldsymbol{v}^{[g]}} \left( \overline{\mathbb{H}}_g \right) < \mathrm{Pr}_{\boldsymbol{v}} \left( \overline{\mathbb{H}} \right) \tag{A.30}$$

**Proof.**

Consider

$$\begin{aligned} &\mathrm{Pr}_{\boldsymbol{v}} \left( \overline{\mathbb{H}} \right) - \prod_{g=1}^{\mathrm{G}} \mathrm{Pr}_{\boldsymbol{v}^{[g]}} \left( \overline{\mathbb{H}}_g \right) \\ &= \mathrm{Pr}_{\boldsymbol{v}} \left( \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \, \|v\| \leq \overline{r} \right\} \setminus \bigcap_{g=1}^{\mathrm{G}} \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \left\| P^{[g]^{\mathsf{T}}} v \right\| \leq \overline{r}_g \right\} \right) \\ &= \mathrm{Pr}_{\boldsymbol{v}} \left( \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \, \|v\| \leq \overline{r} \right\} \setminus \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \forall_{g=1,..,\mathrm{G}} : \left\| P^{[g]^{\mathsf{T}}} v \right\| \leq \overline{r}_g \right\} \right) \\ &= \mathrm{Pr}_{\boldsymbol{v}} \left( \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \|v\| \leq \overline{r}, \, \exists_{g=1,..,\mathrm{G}} : \left\| P^{[g]^{\mathsf{T}}} v \right\| > \overline{r}_g \right\} \right) \\ &= \mathrm{Pr}_{\boldsymbol{v}} \left( \mathbb{A} \right), \text{ with } \mathbb{A} = \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \|v\| \leq \overline{r}, \, \exists_{g=1,..,\mathrm{G}} : \left\| P^{[g]^{\mathsf{T}}} v \right\| > \overline{r}_g \right\} \end{aligned} \tag{A.31}$$

Let $g^*$ be a partition set for which $\overline{r}_{g^*} < \overline{r}$ and

$$\mathbb{B} = \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \|v\| \leq \overline{r}, \, \left\| P^{[g^*]^{\mathsf{T}}} v \right\| > \overline{r}_{g^*} \right\}$$

The principle is to prove that $\Pr_{\boldsymbol{v}}(\mathbb{B}) > 0$, which implies that $\Pr_{\boldsymbol{v}}(\mathbb{A}) > 0$, since $\mathbb{B} \subseteq \mathbb{A}$. Due to $\|v\| = \left\| \sum\limits_{g=1}^{\mathrm{G}} P^{[g]} v^{[g]} \right\|$, set $\mathbb{B}$ can be written as

$$\mathbb{B} = \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \left\| \sum_{g=1}^{\mathrm{G}} P^{[g]} v^{[g]} \right\| \leq \overline{r},\ \left\| {P^{[g^*]}}^{\intercal} v \right\| > \overline{r}_{g^*} \right\} \tag{A.32}$$

From the Minkowski inequality (A.3) follows that $\left\| \sum\limits_{g=1}^{\mathrm{G}} P^{[g]} v^{[g]} \right\| \leq \sum\limits_{g=1}^{\mathrm{G}} \left\| P^{[g]} v^{[g]} \right\|$.
From $\left\| P^{[g]} v^{[g]} \right\| = \left\| {P^{[g]}}^{\intercal} v \right\|$ follows

$$\begin{aligned}
\mathbb{B} &\supseteq \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rcl} \sum\limits_{h=1}^{\mathrm{G}} \left\| {P^{[h]}}^{\intercal} v \right\| & \leq & \overline{r} \\ \left\| {P^{[g^*]}}^{\intercal} v \right\| & > & \overline{r}_{g^*} \end{array} \right\} \\
&= \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rrcl} & \sum\limits_{h=1}^{\mathrm{G}} \left\| {P^{[h]}}^{\intercal} v \right\| & \leq & \overline{r} \\ \overline{r}_{g^*} < & \left\| {P^{[g^*]}}^{\intercal} v \right\| & \leq & \overline{r} \end{array} \right\} \\
&\supseteq \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rrcr} & \sum\limits_{h=1}^{\mathrm{G}} \left\| {P^{[h]}}^{\intercal} v \right\| & \leq & \overline{r} \\ \overline{r}_{g^*} < & \left\| {P^{[g^*]}}^{\intercal} v \right\| & \leq & \frac{1}{2}(\overline{r} + \overline{r}_{g^*}) \end{array} \right\} \\
&\supseteq \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rrcr} & \sum\limits_{h=1}^{\mathrm{G}} \left\| {P^{[h]}}^{\intercal} v \right\| - \left\| {P^{[g^*]}}^{\intercal} v \right\| & \leq & \overline{r} - \frac{1}{2}(\overline{r} + \overline{r}_{g^*}) \\ \overline{r}_{g^*} < & \left\| {P^{[g^*]}}^{\intercal} v \right\| & \leq & \frac{1}{2}(\overline{r} + \overline{r}_{g^*}) \end{array} \right\} \\
&= \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rrcl} & \sum\limits_{h \neq g^*} \left\| {P^{[h]}}^{\intercal} v \right\| & \leq & \frac{1}{2}(\overline{r} - \overline{r}_{g^*}) \\ \overline{r}_{g^*} < & \left\| {P^{[g^*]}}^{\intercal} v \right\| & \leq & \frac{1}{2}(\overline{r} + \overline{r}_{g^*}) \end{array} \right\} \\
&\supseteq \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rrcl} & \forall_{h \neq g^*} \left\| {P^{[h]}}^{\intercal} v \right\| & \leq & \frac{1}{2}\left(\frac{\overline{r} - \overline{r}_{g^*}}{\mathrm{G}-1}\right) \\ \overline{r}_{g^*} < & \left\| {P^{[g^*]}}^{\intercal} v \right\| & \leq & \frac{1}{2}(\overline{r} + \overline{r}_{g^*}) \end{array} \right\} \\
&= \left\{ \sum_{h=1}^{\mathrm{G}} P^{[h]} v^{[h]} = v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rrcl} & \forall_{h \neq g^*} \left\| v^{[h]} \right\| & \leq & \frac{1}{2}\left(\frac{\overline{r} - \overline{r}_{g^*}}{\mathrm{G}-1}\right) \\ \overline{r}_{g^*} < & \left\| v^{[g^*]} \right\| & \leq & \frac{1}{2}(\overline{r} + \overline{r}_{g^*}) \end{array} \right\} = \mathbb{C}
\end{aligned} \tag{A.33}$$

From $\overline{r}_{g^*} < \overline{r} \implies \frac{1}{2}\left(\frac{\overline{r} - \overline{r}_{g^*}}{\mathrm{G}-1}\right) > 0$ for $\mathrm{G} \geq 2$ (i.e. a decomposition exists), follows that there exists a point $v^* \in \mathbb{C} \setminus cl(\mathbb{C})$. Therefore $\mathbb{C}$ is a full subspace of $\mathbb{R}^{\mathrm{N}}$ and

$$\Pr_{\boldsymbol{v}}(\mathbb{A}) \geq \Pr_{\boldsymbol{v}}(\mathbb{B}) \geq \Pr_{\boldsymbol{v}}(\mathbb{C}) > 0$$

■

Consider the definitions of $\underline{\underline{\mathbb{H}}}^{[p]}$ with $p \in \{1, 2, \infty\}$ as in (3.99) which are respectively the largest cube, ball and diamond shape with radius $\underline{r}^{[p]}$ enclosed by the Happy set $\mathbb{H}$. Consider the idea described in Section 4.3.1 of *decomposing* $\underline{\mathbb{H}}^{[p]}$ into lower dimensional sets $\underline{\mathbb{H}}_g^{[p]} \subseteq \mathbb{R}^{\mathrm{M}_g}$, $g = 1,..,\mathrm{G}$. Theorem 4.2 and Corollary 4.6 explain why and when decomposition is effective.

**Theorem 4.2**

Let $\|\cdot\|$ be the abstract notation for a p-norm $\|\cdot\|_p$. Let $\bigcup\limits_{g=1}^{\mathrm{G}} \mathbb{G}_g(x) = \mathbb{S}$ be the decomposition of the uncertain object properties as given in Definition 4.1. Let $(v^{[1]^\intercal},..,v^{[g]^\intercal},..v^{[\mathrm{G}]^\intercal})^\intercal \in \mathbb{R}^{\mathrm{N}}$ be the corresponding decomposition of $v$, where N= $\sum\limits_{g=1}^{\mathrm{G}}\mathrm{M}_g$, $v \in \mathbb{R}^{\mathrm{N}}$ and $v^{[g]} \in \mathbb{R}^{\mathrm{M}_g}$. Let $P^{[g]}$ be a N-by-M$_g$ projection matrix, such that $v = \sum\limits_{g=1}^{\mathrm{G}} P^{[g]}v^{[g]}$, $v^{[g]} = P^{[g]^\intercal}v$ and $\left\|P^{[g]}v^{[g]}\right\| = \left\|v^{[g]}\right\|$. Let $\mathbb{H} = \bigcap\limits_{g=1}^{\mathrm{G}} \left\{v \in \mathbb{R}^{\mathrm{N}} \,\middle|\, P^{[g]^\intercal}v \in \mathbb{H}_g\right\}$, with $\mathbb{H}_g \subseteq \mathbb{R}^{\mathrm{M}_g}$ be the Happy set decomposition. As in (3.98), let $\underline{r} = \min\limits_{v\in\mathbb{H}} \min\limits_{1\le s\le S} \{\|v\| \;\; |u_s(x,v) \in \{\mathrm{L}_s, \mathrm{H}_s\}\}$ define the radius of the largest set $\underline{\mathbb{H}} = \left\{v \in \mathbb{R}^{\mathrm{N}} \,|\|v\| \le \underline{r}\right\}$ enclosed by the Happy set, i.e. $\underline{\mathbb{H}} \subseteq \mathbb{H}$ . Similarly, let $\underline{r}_g = \min\limits_{v^{[g]}\in\mathbb{H}_g} \min\limits_{s\in\mathbb{G}_g} \left\{\left\|v^{[g]}\right\| \;\; \left|u_s(x, P^{[g]}v^{[g]}) \in \{\mathrm{L}_s, \mathrm{H}_s\}\right.\right\}$ define $\underline{\mathbb{H}}_g = \left\{v^{[g]} \in \mathbb{R}^{\mathrm{M}_g} \,\middle|\, \left\|v^{[g]}\right\| \le \underline{r}^{[g]}\right\} \subseteq \mathbb{H}_g$. Let $\boldsymbol{v}$ have independent distributed elements and PDF $f(v) = \prod\limits_{g=1}^{\mathrm{G}} f_g(v^{[g]})$ with $f(v) > 0$ for all $v \in \mathbb{R}^{\mathrm{N}}$. Then

$$\prod_{g=1}^{\mathrm{G}} \mathrm{Pr}_{\boldsymbol{v}^{[g]}}\left(\underline{\mathbb{H}}_g\right) \ge \mathrm{Pr}_{\boldsymbol{v}}\left(\underline{\mathbb{H}}\right) \tag{A.34}$$

**Proof.** The left-hand side of (A.34) can be expressed as

$$\prod_{g=1}^{\mathrm{G}} \mathrm{Pr}_{\boldsymbol{v}^{[g]}}\left(\underline{\mathbb{H}}_g\right) = \mathrm{Pr}_{\boldsymbol{v}}\left(\bigcap_{g=1}^{\mathrm{G}} \left\{v \in \mathbb{R}^{\mathrm{N}} \,\middle|\, P^{[g]^\intercal}v \in \underline{\mathbb{H}}_g\right\}\right) \tag{A.35}$$

It remains to be shown that

$$\bigcap_{g=1}^{\mathrm{G}} \left\{v \in \mathbb{R}^{\mathrm{N}} \,\middle|\, P^{[g]^\intercal}v \in \underline{\mathbb{H}}_g\right\} \supseteq \underline{\mathbb{H}} \tag{A.36}$$

which is equivalent to showing that

$$\bigcap_{g=1}^{\mathrm{G}} \left\{v \in \mathbb{R}^{\mathrm{N}} \,\middle|\, \left\|P^{[g]^\intercal}v\right\| \le \underline{r}_g\right\} \supseteq \left\{v \in \mathbb{R}^{\mathrm{N}} \,|\|v\| \le \underline{r}\right\} \tag{A.37}$$

In (4.7) is explained that $u_s(x,v) = u_s(x, \sum\limits_{g=1}^{\mathrm{G}} P^{[g]}v^{[g]}) = u_s(x, P^{[g]}v^{[g]})$ for $s \in \mathbb{G}_g(x)$, i.e. for a given $x$ the function $u_s$ does not depend on the elements $v^{[k]}$, $k \neq g$. The radius of the largest shape inside $\mathbb{H}_g$ is

$$\begin{aligned}\underline{r}_g &= \min_{v^{[g]}\in\mathbb{H}_g} \ \min_{s\in\mathbb{G}_g} \ \left\{\left\|v^{[g]}\right\| \ \left|u_s(x, P^{[g]}v^{[g]}) \in \{\mathrm{L}_s, \mathrm{H}_s\}\right.\right\} \\ &= \min_{v\in\mathbb{H}} \ \min_{s\in\mathbb{G}_g} \ \{\|v\| \ \ |u_s(x, v) \in \{\mathrm{L}_s, \mathrm{H}_s\}\}\end{aligned} \tag{A.38}$$

because the elements of $v$ corresponding the elements of $v^{[k]}$, $k \neq g$ are free variables and will get the value 0 when minimising the norm. Consequently

$$\begin{aligned}\underline{r} &= \min_{v\in\mathbb{H}} \min_{s\in\mathbb{S}} \{\|v\| \ \ |u_s(x, v) \in \{\mathrm{L}_s, \mathrm{H}_s\}\} \\ &= \min_{g\in\{1,..,\mathrm{G}\}} \underline{r}_g\end{aligned}$$

such that $\underline{r} \leq \underline{r}_g$ for all $g = 1, .., \mathrm{G}$. Finally, for all $v \in \mathbb{R}^{\mathrm{N}}$ with $\|v\| \leq \underline{r}$ holds

$$\left\|P^{[g]^\intercal} v\right\| \leq \underline{r}_g \text{ for any } g = 1, .., \mathrm{G}$$

which proves (A.37) ■

**Corollary 4.6**
If there exists a $g \in \{1, .., \mathrm{G}\}$ with $\underline{r}_g > \underline{r}$ then

$$\prod_{g=1}^{\mathrm{G}} \mathrm{Pr}_{\boldsymbol{v}^{[g]}} \left(\underline{\mathbb{H}}_g\right) > \mathrm{Pr}_{\boldsymbol{v}} \left(\underline{\mathbb{H}}\right) \tag{A.39}$$

**Proof.** Consider

$$\begin{aligned}&\mathrm{Pr}_{\boldsymbol{v}} \left(\underline{\mathbb{H}}\right) - \prod_{g=1}^{\mathrm{G}} \mathrm{Pr}_{\boldsymbol{v}^{[g]}} \left(\underline{\mathbb{H}}_g\right) \\ =&\mathrm{Pr}_{\boldsymbol{v}} \left\{v \in \mathbb{R}^{\mathrm{N}} \left| \forall_{g=1,..,\mathrm{G}} : \left\|P^{[g]^\intercal} v\right\| \leq \underline{r}_g \right.\right\} \setminus \left\{v \in \mathbb{R}^{\mathrm{N}} \left| \|v\| \leq \underline{r}\right.\right\} \\ =&\mathrm{Pr}_{\boldsymbol{v}}(\mathbb{A}) \text{ with } \mathbb{A} = \left\{ v \in \mathbb{R}^{\mathrm{N}} \left| \begin{array}{c} \forall_{g=1,..,\mathrm{G}} : \left\|P^{[g]^\intercal} v\right\| \leq \underline{r}_g \\ \|v\| > \underline{r} \end{array} \right. \right\}\end{aligned} \tag{A.40}$$

The principle is to construct two more sets, $\mathbb{B}$ and $\mathbb{C}$ such that $\mathbb{C} \subseteq \mathbb{B} \subseteq \mathbb{A}$, and prove that $\mathrm{Pr}_{\boldsymbol{v}}(\mathbb{C}) > 0$, which then implies $\mathrm{Pr}_{\boldsymbol{v}}(\mathbb{A}) > 0$. From Lemma A.2 follows that

$$\begin{aligned}\|v\| &= \left\|\left(\left\|v^{[1]}\right\|, \cdots, \left\|v^{[g]}\right\|, \cdots, \left\|v^{[\mathrm{G}]}\right\|\right)\right\| \\ &= \left\|\left(\left\|P^{[1]^\intercal} v\right\|, \cdots, \left\|P^{[g]^\intercal} v\right\|, \cdots, \left\|P^{[\mathrm{G}]\intercal} v\right\|\right)\right\|\end{aligned} \tag{A.41}$$

such that

$$\mathbb{A} \supseteq \left\{ v \in \mathbb{R}^{\mathrm{N}} \left| \begin{array}{c} \forall_{g=1,..,\mathrm{G}} : \left\|P^{[g]^\intercal} v\right\| \leq \underline{r}_g \\ \exists_{g=1,..,\mathrm{G}} : \left\|P^{[g]^\intercal} v\right\| > \underline{r} \end{array} \right. \right\} = \mathbb{B} \tag{A.42}$$

Let $g^*$ be the partition set for which $\underline{r}_{g^*} > \underline{r}$, then

$$\mathbb{B} \supseteq \left\{ v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{c} \forall_{g=1,..,\mathrm{G}} : \left\| P^{[g]^\intercal} v \right\| \leq \underline{r}_g \\ \left\| P^{[g^*]^\intercal} v \right\| > \underline{r} \end{array} \right\} = \mathbb{C} \tag{A.43}$$

Due to $\left\| v^{[g]} \right\| = \left\| P^{[g]^\intercal} v \right\|$, holds

$$\mathbb{C} = \left\{ \sum_{h=1}^{\mathrm{G}} P^{[h]} v^{[h]} = v \in \mathbb{R}^{\mathrm{N}} \;\middle|\; \begin{array}{rr} \forall_{h \neq g^*} : & 0 \leq \\ & \underline{r} < \end{array} \begin{array}{l} \left\| v^{[h]} \right\| \leq \underline{r}_g \\ \left\| v^{[g^*]} \right\| \leq \underline{r}_{g^*} \end{array} \right\} \tag{A.44}$$

From $0 < \overline{r} < \overline{r}_{g^*}$, follows $\mathrm{Pr}_{\boldsymbol{v}}(\mathbb{A}) \geq \mathrm{Pr}_{\boldsymbol{v}}(\mathbb{B}) \geq \mathrm{Pr}_{\boldsymbol{v}}(\mathbb{C}) > 0$ ■

## A.10 Minimum norm on hyper-plane

Let the Hölder norm $\|v\|_p = \left( \sum_{n=1}^{\mathrm{N}} |v_n|^p \right)^{\frac{1}{p}}$ where $p \in (1, \infty)$. The minimum over values of $v$ on a hyper-plane defined by $a \in \mathbb{R}^{\mathrm{N}}$, $b \in \mathbb{R}$ is defined by

$$\min_{v \in \mathbb{R}^{\mathrm{N}}} \left\{ \|v\|_p \,\middle|\, a^\intercal v = b \right\}$$

where $\|a\|_1 > 0$ and $b \neq 0$. Based on the symmetry properties of the p-norm, it follows (without proof) that the minimum norm problem can be expressed as:

$$\min_{v \in \mathbb{R}^{\mathrm{N}}} \left\{ \|v\|_p \,\middle|\, a^\intercal v = b \right\} = \min_{v \in \mathbb{R}^{\mathrm{N}}} \left\{ \|v\|_p \,\middle|\, \sum_{n=1}^{\mathrm{N}} |a_n| \, v_n = |b| \right\}$$

For convention, let $|a|^\intercal v = |b|$ define the so-called *positive hyper-plane*, meaning that all elements of vector $|a|$ and scalar $|b|$ are positive, i.e. $|a| = (|a_1|, .., |a_n|, .., |a_{\mathrm{N}}|)^\intercal$. The inspiration for the concepts underlying Theorem A.4 comes from Hendrix et al. (1996) and Hendrix (1998, page 145)

**Theorem A.4**

$$\min_{v \in \mathbb{R}^{\mathrm{N}}} \left\{ \|v\|_p \,\middle|\, \sum_{n=1}^{\mathrm{N}} |a_n| \, v_n = |b| \right\} = \frac{|b|}{\|a\|_{\frac{p}{p-1}}}$$

**Proof.**

Let $f(v) = \|v\|_p$, then $\min_{v \in \mathbb{R}^{\mathrm{N}}} \left\{ \|v\|_p \,\middle|\, \sum_{n=1}^{\mathrm{N}} |a_n| \, v_n = |b| \right\}$ is equivalent to $\min_{v \in \mathbb{R}^{\mathrm{N}}} f(v)$ subject to $|a|^\intercal \, v = |b|$. The corresponding Karush-Kuhn-Tucker conditions for the minimum point $v$ is: $\exists u \geq 0$ such that $\nabla f(v) = u \, |a|$. This is equivalent to $\exists \alpha \geq 0$ such that

$$\begin{pmatrix} v_1^{p-1} \\ \vdots \\ v_{\mathrm{N}}^{p-1} \end{pmatrix} = \alpha^{p-1} \begin{pmatrix} |a_1| \\ \vdots \\ |a_{\mathrm{N}}| \end{pmatrix} \Longrightarrow \begin{pmatrix} v_1 \\ \vdots \\ v_{\mathrm{N}} \end{pmatrix} = \alpha \begin{pmatrix} |a_1|^{\frac{1}{p-1}} \\ |a_2|^{\frac{1}{p-1}} \\ \vdots \\ |a_{\mathrm{N}}|^{\frac{1}{p-1}} \end{pmatrix}$$

which gives that for the minimum point $v$

$$v_n = \alpha |a_n|^{\frac{1}{p-1}} \text{ for } n = 1, .., \mathrm{N} \tag{A.45}$$

Substitution in $|a|^{\intercal} v = |b|$ gives

$$\alpha \sum_{n=1}^{\mathrm{N}} |a_n|^{\frac{p}{p-1}} = |b| \Longrightarrow \alpha = \frac{|b|}{\sum\limits_{n=1}^{\mathrm{N}} |a_n|^{\frac{p}{p-1}}}$$

Substitution in (A.45) gives that the minimum point is

$$v_n = \frac{|b|}{\sum\limits_{n=1}^{\mathrm{N}} |a_n|^{\frac{p}{p-1}}} |a_n|^{\frac{1}{p-1}} \text{ for } n = 1, .., \mathrm{N} \tag{A.46}$$

The corresponding minimum is

$$\|v\|_p = \frac{|b|}{\sum\limits_{n=1}^{\mathrm{N}} |a_n|^{\frac{p}{p-1}}} \left( \sum_{n=1}^{\mathrm{N}} |a_n|^{\frac{p}{p-1}} \right)^{\frac{1}{p}} = |b| \left( \sum_{n=1}^{\mathrm{N}} |a_n|^{\frac{p}{p-1}} \right)^{\frac{1-p}{p}} = \frac{|b|}{\|a\|^{\frac{p}{p-1}}} \tag{A.47}$$

■

**Corollary A.1** *For $p \downarrow 1$ follows that $\frac{p}{p-1} \to \infty$. Therefore*

$$\min_{v \in R_+^N} \left\{ \|v\|_1 \middle| \sum_{n=1}^{N} |a_n| \, v_n = |b| \right\} = \frac{|b|}{\|a\|_\infty}$$

**Corollary A.2** *For $p \to \infty$ follows that $\frac{p}{p-1} \downarrow 1$. Therefore*

$$\min_{v \in R_+^N} \left\{ \|v\|_\infty \middle| \sum_{n=1}^{N} |a_n| \, v_n = |b| \right\} = \frac{|b|}{\|a\|_1}$$

## A.11 STACO Case: World Regions

The 12 regions in the world, considered in the STACO case are: USA, Japan (JPN), European Economic Community (EEC), other OECD countries (OOC), Eastern European countries (EET), former Soviet Union (FSU), energy exporting countries (EEX), China (CHN), India (IND), Dynamic Asian economies (DAE), Brazil (BRA) and the rest of the world (ROW) Finus et al. (2003).

**Table A.2**: STACO model parameter mean values for three scenarios (Finus et al., 2003)

| $i$ | $\beta_{i,1}$ | | | $\beta_{i,2}$ | $\beta_{i,3}$ |
|---|---|---|---|---|---|
| | 120%-scenario | 200%-scenario | 300%-scenario | | |
| 1 | 10.156 | 16.926 | 25.389 | 0.1715 | 0.0216 |
| 2 | 7.7413 | 12.902 | 19.353 | 7.8270 | 0.6681 |
| 3 | 10.591 | 17.652 | 26.477 | 0.6478 | 0.1034 |
| 4 | 1.5483 | 2.5804 | 3.8706 | 0.0000 | 0.3577 |
| 5 | 0.5834 | 0.9723 | 1.4585 | 0.2095 | 0.3405 |
| 6 | 3.0292 | 5.0487 | 7.5730 | 0.0181 | 0.0991 |
| 7 | 1.3463 | 2.2439 | 3.3658 | 1.3050 | 0.1379 |
| 8 | 2.7824 | 4.6373 | 6.9560 | 0.1030 | 0.0030 |
| 9 | 2.2439 | 3.7397 | 5.6096 | 0.3392 | 0.0647 |
| 10 | 1.1174 | 1.8624 | 2.7936 | 1.6270 | 0.2026 |
| 11 | 0.6866 | 1.1443 | 1.7165 | 36.620 | 24.190 |
| 12 | 3.0517 | 5.0861 | 7.6291 | 0.3470 | 0.0905 |

## A.12 Efficient strategies set in the STACO case

In this section is shown that any cartel coalition strategy $c \in \{0,1\}^n$ as defined in the STACO case (Finus et al., 2003), corresponds almost surely to a unique efficient $CO_2$ reduction strategy $q \in \mathbb{E}$. Also the computation of the efficient $CO_2$ reduction strategies is discussed.

From (5.31) follows that the payoff of a region, only depends linearly on the strategies of the other regions. As a consequence, the efficiency of the choice $q_i$ of region $i$ does not depend on the other ($j \neq i$) strategies $q_j$. To show this, consider two payoff strategy vectors $q \in \mathbb{E}$ and $p \in \mathbb{E}$, with a different value $q_i \neq p_i$ for player $i$, while the other player ($j \neq i$) strategies are equal $q_j = p_j$. Region $i$ can compare these two strategies by taking the difference of, what we called the *payoff aggregation vector* $\Pi(q, v, c)$ in Section 5.5.1, where the model parameters are represented as $v = [\beta_{1,1}, \beta_{1,2}, \beta_{1,3}, \cdots, \beta_{12,1}, \beta_{12,2}, \beta_{12,3}]^\intercal$. The model parameter vector $v$ has N = 36 elements. In the situation that region $i$ is not a member of the coalition we have:

$$\begin{aligned}&\Pi_i(q, v, c) - \Pi_i(p, v, c) = \\ &\beta_{i,1}(q_i - p_i) - \frac{1}{2}\beta_{i,2}(q_i^2 - p_i^2) - \frac{1}{3}\beta_{i,3}(q_i^3 - p_i^3)\end{aligned} \tag{A.48}$$

In the situation that region $i$ is a member of the coalition we have:

$$\begin{aligned}&\Pi_i(q, v, c) - \Pi_i(p, v, c) = \\ &\sum_{j \in \mathbb{C}(c)} \beta_{j,1}(q_i - p_i) - \frac{1}{2}\beta_{i,2}(q_i^2 - p_i^2) - \frac{1}{3}\beta_{i,3}(q_i^3 - p_i^3)\end{aligned} \tag{A.49}$$

Both (A.48) and (A.49) are independent of $q_j$ for $j \neq i$. Region $i$ can judge which of its individual choices, $q_i$ or $p_i$ will yield a higher (aggregate) payoff, without having to know the other region strategies. In particular, now we know that region $i$ is capable to select the optimal strategy $q_i^*$ from a *finite* set of strategies, without knowing $q_j$ for $j \neq i$.

Since the payoff efficiency of region $i$ only depends on $q_i$, we can conclude that the most efficient choice for region $i$ must be either located at one of the *boundaries*[4], i.e. $\min(\mathbb{E}_i)$ or $\max(\mathbb{E}_i)$, or anywhere in the interior of $\mathbb{E}_i$ where the first derivative with respect to $q_i$ is zero:

$$\frac{\partial \mathit{\Pi}_i(q, v, c)}{\partial q_i} = \left( c_i \sum_{j \in \mathbb{C}(c) \setminus \{i\}} \beta_{j,1} \right) + \beta_{i,1} - \beta_{i,2} q_i - \beta_{i,3} q_i^2 = 0 \tag{A.50}$$

which can be found with the ABC-formula. Since the dimension of the set

$$\left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \left( c_i \sum_{j \in \mathbb{C}(c) \setminus \{i\}} \beta_{j,1} \right) + \beta_{i,1} = 0, \beta_{i,2} = 0, \beta_{i,3} = 0 \right\} \tag{A.51}$$

is lower than N = 36, its probability mass is 0 and consequently, for a continuous random vector $\boldsymbol{v}$, (A.50) almost surely (a.s.) has at most two solutions. Together with the vertex solutions, region $i$ can distinguish a.s. at most 4 *candidate efficient strategies* of which the strategy corresponding to the highest payoff can be selected. We conclude that a cartel coalition strategy $c \in \{0, 1\}^n$ and a continuous random vector $\boldsymbol{v}$, a.s. determines a unique[5] efficient $CO_2$ reduction strategy in the STACO case.

In the following, the relation is elaborated between the payoff difference function (5.32) and candidate efficient strategies and under which condition the payoff difference function is continuous in $v$. The four candidate efficient strategies for region $i$ and strategy $c$ are

$$\begin{aligned}
q_i^{[1]}(c, v) &= \min(\mathbb{E}_i) \\
q_i^{[2]}(c, v) &= \max(\mathbb{E}_i) \\
q_i^{[3]}(c, v) &= \min\left( q_i^{[2]}(c, v), \max(q_i^{[1]}(c, v), t_i^{[1]}(c, v)) \right) \\
q_i^{[4]}(c, v) &= \min\left( q_i^{[2]}(c, v), \max(q_i^{[1]}(c, v), t_i^{[2]}(c, v)) \right)
\end{aligned} \tag{A.52}$$

[4] Lower bound: In Finus et al. (2003) it is argued that negative strategies are irrational. Upper bound: A player can at most reduce as much $CO_2$ as the player is emitting.

[5] Theoretically the event that there are two or more candidate efficient strategy solutions, corresponding to exactly the same maximal payoff value can occur. From (A.56) follows that this situation is irrelevant.

with

$$t_i^{[1]}(c,v) = \begin{cases} \frac{\beta_{i,2}-\sqrt{\beta_{i,2}^2+4\beta_{i,3}\gamma_i(c,v)}}{2\beta_{i,3}} & \text{if } \beta_{i,2}^2+4\beta_{i,3}\gamma_i(c,v)\geq 0 \text{ and } \beta_{i,3}\neq 0 \\ \frac{\gamma_i(c,v)}{\beta_{i,2}} & \text{if } \beta_{i,3}=0 \text{ and } \beta_{i,2}\neq 0 \\ -\infty & \text{elsewhere} \end{cases}$$

$$t_i^{[2]}(c,v) = \begin{cases} \frac{\beta_{i,2}+\sqrt{\beta_{i,2}^2+4\beta_{i,3}\gamma_i(c,v)}}{2\beta_{i,3}} & \text{if } \beta_{i,2}^2+4\beta_{i,3}\gamma_i(c,v)\geq 0 \text{ and } \beta_{i,3}\neq 0 \\ \frac{\gamma_i(c,v)}{\beta_{i,2}} & \text{if } \beta_{i,3}=0 \text{ and } \beta_{i,2}\neq 0 \\ \infty & \text{elsewhere} \end{cases}$$

$$\gamma_i(c,v) = \left( c_i \sum_{j\in\mathbb{C}(c)\setminus\{i\}} \beta_{j,1} \right) + \beta_{i,1} \tag{A.53}$$

It can be shown that the functions $q_i^{[1]}(c,v)$, $q_i^{[2]}(c,v)$, $q_i^{[3]}(c,v)$ and $q_i^{[3]}(c,v)$ are continuous in each point $v$ where $\beta_{i,2}^2+4\beta_{i,3}\gamma_i(c,v)\neq 0$, $\beta_{i,2}\neq 0$ and $\beta_{i,3}\neq 0$ with $i=1,..,12$. Consequently the function

$$\max\left\{\pi_k(q^*,v)\,\middle|\, q_i^*\in\left\{q_i^{[1]}(c,v),q_i^{[2]}(c,v),q_i^{[3]}(c,v),q_i^{[4]}(c,v)\right\} \text{ for } i=1,..,12\right\} \tag{A.54}$$

with $k=1,..,12$, is continuous in each point $v$ where $\beta_{i,2}^2+4\beta_{i,3}\gamma_i(c,v)\neq 0$, $\beta_{i,2}\neq 0$ and $\beta_{i,3}\neq 0$ with $i=1,..,12$. Finally, the payoff difference function (5.32) can be expressed as

$$u_k(c,v) =$$

$$\max\left\{\pi_k(q^*,v)\,\middle|\, q_i^*\in\left\{q_i^{[j]}(c,v)\right\} \text{ for } i=1,..,12,\ j=1,2,3,4\right\} \tag{A.55}$$

$$-$$

$$\max\left\{\pi_k(q^{**},v)\,\middle|\, q_i^{**}\in\left\{q_i^{[j]}(nc(c,i),v)\right\} \text{ for } i=1,..,12,\ j=1,2,3,4\right\} \tag{A.56}$$

with $k=1,..,12$, and leads to the conclusion that the payoff difference function is continuous in each point $v$ where $\beta_{i,2}^2+4\beta_{i,3}\gamma_i(c,v)\neq 0$, $\beta_{i,2}^2+4\beta_{i,3}\gamma_i(nc(c,i),v)\neq 0$, $\beta_{i,2}\neq 0$ and $\beta_{i,3}\neq 0$ with $i=1,..,12$.

## Robustness Programming

Robustness of an object is defined as the probability that an object will have properties as required. Robustness Programming (RP) is a mathematical approach for Robustness estimation and Robustness optimisation. Robustness Programming is an extension of Stochastic Programming and is intended to support decision making with respect to uncertainty. For example, consider the investment in a portfolio of stock market shares. The number of shares to invest in, is typically a controllable factor, but the future price of shares and the resulting portfolio return are uncontrollable factors. It is interesting to find the combination of shares that maximises the probability of receiving some predefined return target. An example in the context of designing a food product, is to find the best mixture of ingredients, such that a product is appealing but also safe. The mixture of ingredients can be assumed a controllable factor, whereas bacteria growth rate affecting temperature fluctuations, between production and consumption, are uncontrollable. It is interesting to find a mixture design of ingredients, with maximal probability that the number of bacteria at the moment of consumption is below safety limits.

## Robustness Programming Framework

The first research question deals with finding a generic notation for RP problems and RP solution methods and resulted in the Robustness Programming Framework as presented in Chapter 2. The RP framework is based on the assumption that object properties can be modelled as a continuous function $u(x, v)$, where vector $x \in \mathbb{R}^{\mathrm{I}}$ represents the values of I controllable factors and $v \in \mathbb{R}^{\mathrm{N}}$ represents N uncontrollable factors. The object properties design objective is $\mathrm{L} \leq u(x, v) \leq \mathrm{H}$, i.e. if the object properties are between the bounds L and H, then the object properties are as required. All uncontrollable factor values, for which the object properties are as required, are in the so-called Happy set: $\mathbb{H}(x) = \left\{v \in \mathbb{R}^{\mathrm{N}} | \mathrm{L} \leq u(x, v) \leq \mathrm{H}\right\}$ and Robustness is defined as $R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$, given the probability distribution of the random uncontrollable factors $\boldsymbol{v}$.

The second research question is 'What are the characteristics of an RP problem, that provide sufficient information to decide about applicable RP solution methods?'. The probability distribution of $\boldsymbol{v}$ and geometric properties of $\mathbb{H}(x)$ play a central role in answering this question. Table 6.1 gives an overview of the sufficient conditions found for applicability of RP methods, when given the RP problem properties.

## Robustness Programming methods

In the general case, $R(x)$ cannot be determined analytically and is estimated instead. The Monte Carlo (MC) method is seen as a standard approach to estimate $R(x)$. However, the MC estimate function of $R(x)$ is a discontinuous step function, generally with infinitely many local optima and has therefore bad optimisation characteristics. This observation triggered the research for alternative RP methods. The following RP methods are investigated in Chapters 3 and 4:

- The Monte Carlo (MC) sampling method. The MC estimator of $R(x)$ is unbiased.

- The Smoothed Monte Carlo (SMC) sampling method. The SMC method is a smoothed version of the MC method, such that the SMC estimate function is continuous a.e. in $\mathbb{X}$. The SMC method deviates at most $\frac{1}{2\mathrm{M}}$ from the MC estimate, where M is the number of samples. This means that the SMC estimate can be made arbitrarily close to the unbiased MC estimate, by choosing M large enough.

- The N-1MC sampling method. The N-1MC estimator of $R(x)$ is unbiased. The method is based on sampling N-1 elements of $\boldsymbol{v}$ with the Monte Carlo method and using the CDF of the one remaining element of $\boldsymbol{v}$ to determine the N-1MC estimate.

- The Directional Sampling (DS) method. The DS estimator of $R(x)$ is unbiased. The DS method is applicable if $\boldsymbol{v}$ has a so-called spherical symmetric distribution, like the Normal distribution. The DS estimate determines the length of rays in the Happy set, through uniformly distributed points on the unit sphere, to compute the unbiased Robustness estimate.

- The Exponential Simplex (ES) sampling method. The ES estimator of $R(x)$ is unbiased. The ES method is applicable if $\boldsymbol{v}$ is (two-sided) Exponentially distributed. The ES estimate uses the length of rays in the Happy set, through uniformly distributed points on the unit simplex.

- The Diamond, Ball and Cube methods are bounding methods, based on the geometric shape of $\mathbb{B}^{[p]}(r) = \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \|v\|_p \leq r \right\}$ defined by the p-norm with $p = 1, 2, \infty$ respectively. The most effective Robustness lower bound $\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[p]}(r^*)\right\}$ is determined by $r^* = \max\left\{r|\mathbb{B}^{[p]}(r) \subseteq \mathbb{H}(x)\right\}$ and the most effective upper bound is determined by $r^* = \min\left\{r|\mathbb{B}^{[p]}(r) \supseteq \mathbb{H}(x)\right\}$. The idea is that the geometric shape of the bounding sets, makes it easier to compute the Robustness bound than the actual Robustness $R(x)$.

- The Compression method and Decomposition method are computational methods to make the Robustness estimation methods more efficient. Both methods are based on the principle that the efficiency difference between the MC method and the alternative estimation methods, becomes smaller for increasing Happy set dimension N. Hence, compressing the Happy set or decomposing the Happy set into lower dimensional Happy sets can improve the efficiency of the estimation methods.

- The Warm Start (WS) method is a method to let an iterative optimisation algorithm, detect a Robustness improvement direction, even if the estimate of $R(x)$ is zero in a small neighbourhood around $x$.

The MC method is used as a reference RP method for comparing the alternative RP methods. The DS method is a generalisation of the methods developed by István Deák (2000, 2003). All other methods were developed during this PhD research.

The third research question deals with identifying mathematical properties of RP methods, that give information about the quality of RP methods, relevant for Robustness estimation and Robustness optimisation. Effectiveness, efficiency and applicability of RP methods are relevant for comparing the quality of RP methods.

- *Effectiveness:* In general, Robustness $R(x)$ cannot be computed directly and is estimated instead. The objective is to estimate $R(x)$ effectively. An RP method is defined effective if the RP method is unbiased, since that would mean $R(x)$ can be estimated arbitrarily accurate, given sufficient samples.

  The MC, N-1MC, DS and ES method are effective estimation methods because they are all based on unbiased Robustness estimators. The SMC method deviates at most $\frac{1}{2\mathrm{M}}$ from the MC estimate, where M is the number of samples. This means that the SMC estimate can be made arbitrarily close to the unbiased MC estimate, by choosing M large enough.

  The effectiveness of the Diamond, Ball and Cube bounding methods depends on the RP problem. To identify the most effective bound involves solving a Mathematical Programming (MP) problem. Only if the global solution is found for the corresponding MP problems, the bound is correct and optimally effective.

  The Compression and Decomposition method do not change the unbiasedness of estimation methods. Compression leads to more effective bounds and it is found out under which conditions decomposition leads to strictly more effective bounds.

- *Efficiency:* The efficiency of Robustness estimation methods, that are based on sampling, is expressed as the standard error of the estimator given an M=1 sample. The N-1MC, DS and ES methods are strictly more efficient than the MC method, only if the Happy set does not have an All-Or-Nothing shape, DS-Radial-Shape or ES-radial-shape, respectively. Performance indicators are defined to measure the Robustness optimisation efficiency of the RP methods. Statistical inference on these performance indicators enables to identify the RP methods that perform significantly above average.

- *Applicability:* The mathematical properties of the RP methods are relevant to identify the appropriate Robustness estimation and computation methods, when given an RP problem. Table 6.1 gives an overview of RP problem characteristics, regarding the information about uncontrollable factors $\boldsymbol{v}$ and the structure of the Happy set $\mathbb{H}(x)$, for which the mentioned RP methods are applicable.

### Case studies

The fourth research question is how to compare the performance of RP methods. The performance of Robustness estimation methods is assessed by comparing the number of samples that are required to reach a predefined standard error level. The best performing Robustness estimation method, is the method that requires the least number of samples to reach the predefined standard error level. The performance of Robustness optimisation methods is assessed with defined performance indicators. These performance indicators are based on identifying the RP methods that result in the highest Robustness value found after a limited number of optimisation iterations. For practical reasons we choose 20 optimisation iterations as the limit.

It is concluded that the computation time (in sec.) and number of floating-point operations (in FLOPS) are not suitable for an objective performance comparison, because the computation time and arithmetic efficiency depends on software quality, computer memory and processor speed.

Two case studies have been carried out for Unilever R&D: One dealing with Robust mixture designs and one dealing with Robust material cost planning. Additionally, a case study is derived from the Robust mixture design case, to assess the performance of the ES method. The Happy sets in the Unilever cases all possess a polyhedral structure. From the Unilever case studies the following can be concluded:

- The Warm Start method is efficient in combination with the MC, N-1MC, DS and ES method.
- Decomposition and Compression are efficient for all cases. However, Decomposition and Compression require extra computations and let the computation time increase.
- The efficiency of the SMC, N-1MC, DS and ES methods, is above average for at least one of the investigated Unilever cases.
- For all Unilever cases, the MC, Ball, Diamond and Cube method are not efficient.

One case study was done in an Environmental Economics research program called STACO at Wageningen University and deals with the Robustness of coalitions of world regions for jointly reducing $CO_2$ emissions. From the STACO case study the following can be concluded: The STACO case illustrates that the DS method is more efficient per sample than the MC method, for the majority of the assessed scenarios. From a computational point of view, the measurements support the conclusion that the MC implementation is more efficient than the DS implementation.

## Robuustheid Programmering

De Robuustheid van een object is gedefinieerd als de kans dat eigenschappen van een object aan gestelde eisen zullen voldoen. Robuustheid Programmering (RP) is een wiskundige aanpak voor het schatten en optimaliseren van Robuustheid. RP is een uitbreiding van Stochastische Programmering en is bedoeld voor het ondersteunen van besluitvorming onder onzekerheid. In RP worden controleerbare factoren typisch als beslissingsvariabelen gemodelleerd en worden oncontroleerbare factoren als kansvariabelen gemodelleerd.

Neem bijvoorbeeld de investering in een portfolio van aandelen. Het aantal aandelen waarin geïnvesteerd wordt is typisch een controleerbare factor, terwijl de toekomstige aandelenprijzen en de resulterende portfolio-opbrengst juist oncontroleerbare factoren zijn. Het is interessant de samenstelling van aandelen te vinden waarvoor de kans maximaal is op het behalen van een vooraf vastgestelde opbrengstdoelstelling.

Een voorbeeld in de context van het ontwerpen van voedingsmiddelen is het vinden van de beste samenstelling van ingrediënten, zodat het resultaat zowel aantrekkelijk als maximaal veilig is. De samenstelling van ingrediënten kan als een controleerbare factor worden beschouwd. De temperatuurverandering in de tijd tussen productie en consumptie beïnvloedt bacteriële groeisnelheden en is een voorbeeld van een oncontroleerbare factor. Het is interessant om een samenstelling van ingrediënten te vinden, waarbij de kans zo groot mogelijk is dat het aantal bacteria op het moment van consumptie voldoet aan gestelde veiligheidseisen.

## Raamwerk van Robuustheid Programmering

De eerste onderzoeksvraag betreft het vinden van een generieke notatie voor RP-problemen en RP-oplossingsmethoden en heeft geresulteerd in het raamwerk van Robuustheid Programmering zoals verwoord in Hoofdstuk 2.

Het raamwerk van RP is gebaseerd op de aanname dat objecteigenschappen gemodelleerd kunnen worden als een continue functie $u(x, v)$, waarbij de vector $x \in \mathbb{R}^{\mathrm{I}}$ de waarde vertegenwoordigt van I controleerbare factoren en $v \in \mathbb{R}^{\mathrm{N}}$ de waarde vertegenwoordigt van N oncontroleerbare factoren. De ontwerpdoelstelling van de objecteigenschappen is $\mathrm{L} \leq u(x, v) \leq \mathrm{H}$, i.e. als de objecteigenschappen tussen de grenzen L en H liggen, dan voldoen de objecteigenschappen aan de gestelde eisen. De zogenaamde Happy set is de verzameling van alle mogelijke waarden voor de oncontroleerbare factoren, waarvoor geldt dat de objecteigenschappen aan de gestelde eisen voldoen: $\mathbb{H}(x) = \left\{v \in \mathbb{R}^{\mathrm{N}} | \mathrm{L} \leq u(x, v) \leq \mathrm{H}\right\}$. Robuustheid is gedefinieerd als de kans $R(x) = \Pr\{\boldsymbol{v} \in \mathbb{H}(x)\}$, gegeven een ontwerp $x$ en de kansverdeling van de stochastische oncontroleerbare factor $\boldsymbol{v}$.

De tweede onderzoeksvraag is 'Wat zijn de karakteristieken van een RP-probleem die voldoende informatie verschaffen om te bepalen welke RP-oplossingsmethoden toepasbaar zijn?' De kansverdeling van $\boldsymbol{v}$ en geometrische eigenschappen van $\mathbb{H}(x)$ zijn de RP probleem eigenschappen die centraal staan in de beantwoording van deze vraag. In Tabel 6.1 staat een overzicht van condities die voldoende zijn voor de toepasbaarheid van de betreffende RP-methoden, gegeven de RP-probleemeigenschappen.

### RP-methoden

In het algemene geval kan $R(x)$ niet analytisch worden bepaald. In plaats daarvan wordt $R(x)$ geschat. De Monte Carlo-methode (MC) wordt gezien als de standaard methode voor het schatten van $R(x)$. Echter de MC-schattingsfunctie van $R(x)$ is een discontinue stapfunctie, met in het algemeen oneindig veel lokale optima en heeft daarom slechte optimaliseringseigenschappen. Deze observatie stimuleerde het onderzoek naar alternatieve RP-methoden. De volgende RP-methoden zijn onderzocht in de Hoofdstukken 3 en 4:

- De Monte Carlo-samplingmethode (MC). De MC-schatter van $R(x)$ is een zuivere schatter.

- De Smoothed Monte Carlo-samplingmethode (SMC). De SMC-methode is een geeffende versie van de MC-methode, zodanig dat de SMC-schattingsfunctie vrijwel overal continu is in $\mathbb{X}$. De SMC-methode wijkt hoogstens $\frac{1}{2\mathrm{M}}$ af van de MC-schatting, met M het aantal trekkingen.

- De N$-$1MC-samplingmethode. De N$-$1MC-schatter van $R(x)$ is een zuivere schatter. De methode bestaat in hoofdlijnen uit twee stappen. In de eerste stap wordt één trekking gedaan van N$-$1 elementen van $\boldsymbol{v}$. In de tweede stap wordt de Robuustheid geschat op basis van de CDF van het overgebleven element van $\boldsymbol{v}$.

- De Directional Samplingmethode (DS). De DS schatter van $R(x)$ is een zuivere schatter. De DS-methode is toepasbaar als $\boldsymbol{v}$ een zogenaamde bolsymmetrische kansverdeling heeft, zoals het geval is bij de standaard Normale kansverdeling. De DS-schatting is gebaseerd op de lengte van lijnstukken in de Happy set. Deze lijnstukken starten in de oorsprong en gaan door uniform gegenereerde punten op de eenheidcirkel.

- De Exponentiële Simplex-samplingmethode (ES). De ES-schatter van $R(x)$ is een zuivere schatter. De ES-methode is toepasbaar als $\boldsymbol{v}$ een (tweezijdige) Exponentiële kansverdeling heeft. De ES-schatting is gebaseerd op de lengte van lijnstukken in de Happy set. Deze lijnstukken starten in de oorsprong en gaan door uniform gegenereerde punten op het eenheidsimplex.

- De Diamond-, Ball- en Cube-schattingsmethode kunnen worden gebruikt voor het bepalen van onder- en bovengrenzen voor de Robuustheid $R(x)$ en zijn gebaseerd op de geometrische vorm van bol $\mathbb{B}^{[p]}(r) = \left\{ v \in \mathbb{R}^{\mathrm{N}} \middle| \|v\|_p \leq r \right\}$ voor respectievelijk $p = 1, 2$ en $\infty$. De Robuustheidgrens is $\Pr\left\{\boldsymbol{v} \in \mathbb{B}^{[p]}(r^*)\right\}$, waarbij de meest effectieve Robuustheidondergrens is gedefinieerd door $r^* = \max\left\{r | \mathbb{B}^{[p]}(r) \subseteq \mathbb{H}(x)\right\}$ en de meest effectieve Robuustheidbovengrens is gedefinieerd door
$r^* = \min\left\{r | \mathbb{B}^{[p]}(r) \supseteq \mathbb{H}(x)\right\}$. De gedachte achter deze methoden is dat de kansmassa van $\mathbb{B}^{[p]}(r)$ relatief eenvoudig kan worden bepaald in vergelijking met de kansmassa van de Happy set, omdat $\mathbb{B}^{[p]}(r)$ daarvoor een geschikte geometrische vorm heeft.

- De Compressie-methode en Decompositie-methode zijn beide zogenaamde dimensie-reductiemethoden die de Robuustheidschattingsmethoden meer efficiënt maken. Dimensie-reductiemethoden zijn gebaseerd op het principe dat het efficiëntieverschil tussen de MC-methode en de overige schattingsmethoden afneemt als de dimensie N van de Happy set toeneemt. Dit betekent dat compressie van de Happy set of decompositie van de Happy set naar lager dimensionale Happy sets, de efficiëntie van de schattingsmethoden kan doen verbeteren ten opzichte van de MC-methode.

- De Warm Start-methode (WS). In het geval de schatting van Robuustheid $R(x)$ nul is in een kleine omgeving rond het punt $x$, dan kan met de WS-methode een punt gegenereerd worden met een positieve schatting van de robuustheid. Het idee van WS is vergelijkbaar met de eerste fase van de twee-fasen-methode bij Lineaire Programmering.

De MC-methode is gebruikt als een referentie RP-methode voor het vergelijken van de alternatieve RP-methoden. De DS-methode is een generalisatie van methoden die zijn ontwikkeld door István Deák (2000, 2003). Alle overige methoden zijn gedurende dit onderzoek ontwikkeld.

De derde onderzoeksvraag betreft het identificeren van wiskundige eigenschappen van RP-methoden, die informatie geven over de kwaliteit van de RP-methoden die relevant zijn voor Robuustheid schatting en Robuustheid optimalisering.

- *Effectiviteit:* In het algemene geval kan Robuustheid $R(x)$ niet rechtstreeks berekend worden, maar wordt er een schatting gemaakt. De doelstelling is om $R(x)$ op een effectieve manier te schatten. Een RP-methode wordt 'effectief' genoemd als de RP-methode een zuivere schatting oplevert. Immers, in die situatie kan een willekeurig nauwkeurige Robuustheid schatting verkregen worden door voldoende trekkingen te gebruiken.

  De MC-, N−1MC-, DS- en ES-methode zijn effectieve schattingsmethoden, omdat ze op een zuivere schatter zijn gebaseerd. De SMC-method wijkt hooguit $\frac{1}{2\mathrm{M}}$ af van de MC schatting, waarbij M het aantal trekkingen is. Dit betekent dat het verschil tussen de SMC-schatting en de MC-schatting willekeurig klein gemaakt kan worden door het aantal trekkingen M groot genoeg te kiezen.

  De effectiviteit van de Diamond, Ball and Cube begrenzingsmethode hangt af van het RP-probleem. Een Mathematisch Programmeringsprobleem (MP) dient te worden opgelost om de meest effectieve Robuustheidgrens te vinden. Het is noodzakelijk de globale oplossing van het MP-probleem te vinden, omdat alleen deze correspondeert met een correcte Robuustheidgrens.

  De Compressie-methode en Decompositie-methode hebben geen invloed op de zuiverheid van schatters in RP-methoden die op sampling zijn gebaseerd. Compressie geeft in de regel wel een meer effectieve Robuustheidgrens. In dit onderzoek is de conditie bepaald waarbij Compressie leidt tot een strikte verbetering van de effectiviteit van de Diamond-, Ball- en Cube- schattingsmethode.

- *Efficiëntie:* De efficiëntie van Robuustheidschattingsmethoden die op sampling zijn gebaseerd, is uitgedrukt als de standard error van de schatter voor M=1 trekkingen.

De N$-$1MC-, DS- en ES- methode zijn efficiënter dan de MC-methode als de Happy set respectievelijk geen All-Or-Nothing shape, DS-Radial-Shape of ES-radial-shape heeft. Prestatie-indicatoren zijn gedefinieerd om de efficiëntie te meten van RP-methoden in de context van Robuustheidoptimalisatie. Statistische gevolgtrekking op basis van deze prestatie indicatoren maakt het mogelijk om vast te stellen welke RP-methoden significant bovengemiddeld presteren.

- *Toepasbaarheid:* Een aantal wiskundige eigenschappen van de RP-methoden zijn relevant voor het bepalen van geschikte Robuustheidschattingsmethoden en -berekeningsmethoden, gegeven een bepaald RP-probleem. Tabel 6.1 geeft een overzicht van RP-probleemeigenschappen, betreffende oncontroleerbare factoren $\boldsymbol{v}$ en de structuur van de Happy set $\mathbb{H}(x)$, waarvoor de genoemde RP-methoden geschikt zijn.

### Casestudy's

De vierde onderzoeksvraag betreft de vraag hoe de prestaties van de RP-methoden kunnen worden vergeleken. De prestaties van Robuustheidschattingsmethoden zijn onderzocht door te vergelijken hoeveel trekkingen nodig zijn om een schatting met een zekere nauwkeurigheid te bereiken. De best presterende Robuustheid schattingsmethode is de methode die het kleinste aantal trekkingen nodig heeft. De prestatie van RP-methoden in de context van Robuustheidoptimalisatie is gemeten aan de hand van prestatie-indicatoren die in dit onderzoek zijn gedefinieerd. Deze prestatie-indicatoren zijn gebaseerd op het bepalen van de RP-methode waarmee het optimaliseringsalgoritme de hoogste waarde van Robuustheid vindt na een beperkt aantal iteraties. Om praktische redenen is voor 20 optimalisatie iteraties gekozen.

Tijdens dit onderzoek is besloten dat de berekeningstijd (in sec.) en het aantal floating-point operations (in FLOPS) niet geschikt zijn voor een objectieve prestatievergelijking, omdat de rekentijd en berekeningsefficiëntie afhangen van het computergeheugen, de processorsnelheid en de kwaliteit van de programmatuur.

Twee casestudy's zijn gedaan voor Unilever R&D. De eerste case betreft het ontwerpen van samenstellingen van ingrediënten die leidt tot een product met maximale Robuustheid ten aanzien van de productspecificaties. De tweede case betreft Robuuste planning van grondstofkosten. Op basis van de eerste case is een numerieke studie geformuleerd om de prestaties van de ES-methode te onderzoeken. Alle Happy sets in de Unilever casestudy's hebben een polyhedrale structuur. De volgende conclusies zijn op de Unilever casestudy's gebaseerd:

- De Warm Start methode is efficiënt in combinatie met de MC-, N$-$1MC-, DS- en ES-methode.
- Decompositie and Compressie zijn efficiënt in alle bestudeerde gevallen. Echter, Decompositie en Compressie vereisen extra berekeningen en laten de rekentijd toenemen.
- De efficiëntie van de SMC-, N$-$1MC-, DS- en ES-methode is bovengemiddeld voor tenminste één van de onderzochte Unilever casestudy's.

- Voor alle Unilever case study's zijn de MC-, Ball-, Diamond- en Cube-methode niet efficiënt.

Eén casestudy is uitgevoerd binnen een milieu-economisch onderzoeksprogramma van Wageningen Universiteit, genaamd STACO. Deze casestudy gaat over de Robuustheid van coalities van wereldregio's die gezamenlijk $CO_2$-emissies reduceren. De volgende conclusies kunnen worden getrokken op basis van de STACO-casestudy: De STACO-case laat zien dat voor de meerderheid van de onderzochte scenario's de DS-methode efficiënter is per trekking dan de MC-methode. Vanuit een rekentechnisch standpunt kan worden gesteld dat de metingen aantonen dat de MC-implementatie efficiënter is dan de DS-implementatie.

## ABOUT THE AUTHOR

Niels Olieman (1973) was born in Leidschendam, The Netherlands. He completed his secondary education at the 'Christelijke Scholengemeenschap Het Loo', Voorburg, in 1992. He studied at Wageningen University from 1992 until 1998 and specialised in Operations Research and Systems Management. He obtained his MSc degree in Agricultural Systems Science and wrote two MSc theses. The first thesis is an Operations Research study at Philips Center For Manufacturing Technology (CFT) in Eindhoven and deals with reliability optimisation. The second thesis is a Systems Engineering study at Wageningen University and deals with re-usable software components for supply chain modelling. During his practical period he developed an inventory management system for a greenhouse construction company called Prins Greenhouses, located in Abbotsford, B.C., Canada.

After his graduation and a brief consultancy job at Prins Greenhouses, he worked for four years as a software engineer for IMN N.V. located in Zeist, the Netherlands. During this time he specialised in Data-warehouse application development.

In April 2002 he started his PhD research at the Operations Research and Logistics group of Wageningen University and worked part-time for the Representative Advisory Council of Wageningen University.

Currently he is working for the Group Risk Management department of Rabobank Netherlands and is member of the team responsible for the validation and review of credit rating models and assuring BISII and DNB compliance.

## TRAINING AND SUPERVISION PLAN

| *Description* | *Institute/Location* | *Year* | *Credits*[1] |
|---|---|---|---|
| **General courses:** | | | |
| Uncertainty Analysis (S02) | SENSE[2] | 2004 | 2.0 |
| Writing and Presenting a Scientific Paper | Mansholt Graduate School | 2002 | 1.0 |
| **Multi-disciplinary courses/activities:** | | | |
| Mansholt Multidisciplinary Seminar | Mansholt Graduate School | 2006 | 1.0 |
| Mansholt Introduction course | Mansholt Graduate School | 2002 | 1.0 |
| **Discipline-specific courses:** | | | |
| Stochastic Programming and Randomised Algorithms | LNMB[3] | 2003 | 4.5 |
| Applications of Game Theory | LNMB | 2003 | 2.3 |
| Game Theory | LNMB | 2003 | 2.8 |
| Simulation | LNMB | 2002 | 1.5 |
| Convex Analysis for Optimisation | LNMB | 2002 | 0.5 |
| Convex Optimisation | LNMB | 2002 | 2.8 |
| Operations Research and Logistics | LNMB | 2002 | 2.8 |
| Combinatorial Optimisation I | LNMB | 2001 | 1.0 |
| **Presentations at international conferences:** | | | 2.0 |
| International Workshop on Global Optimisation | Almería, Spain | 2005 | |
| Operations Research 2004 International Conference | Tilburg, The Netherlands | 2004 | |
| Twenty-Ninth Conference on the Mathematics of Operations Research | Lunteren, The Netherlands | 2004 | |
| The 18th International Symposium on Mathematical Programming | Copenhagen, Denmark | 2003 | |
| Twenty-Eighth Conference on the Mathematics of Operations Research | Lunteren, The Netherlands | 2003 | |
| **Teaching activities:** | | | |
| ORL20304 Practicum Besliskunde I | Wageningen Universiteit | 2005 | |
| Thesis supervision of student on Optimal Robust Product Design | Unilever R&D | 2004 | |
| Thesis supervision of student on Optimal Product Design | Unilever R&D | 2003 | |
| **Total** (min. 20 Credits) | | | **25.2** |

[1] One Credit represents (on average) 40 hours of work

[2] Research School for Socio-Economic and Natural Sciences of the Environment

[3] Dutch Network on the Mathematics of Operations Research