Solving probabilistic-robust optimization problems using methods from semi-infinite optimization

Vom Fachbereich Mathematik der
Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades

Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)

genehmigte Dissertation

von
Holger Marten Berthold

Gutachter: Prof. Dr. Karl-Heinz Küfer
PD Dr. René Henrion
Prof. Dr. Oliver Stein

Datum der Disputation: 22. Juli 2022

DE-386

Danksagung

Vor Beginn des fachlichen Teils möchte ich mich bei allen bedanken, die zum Gelingen dieser Arbeit beigetragen haben.
Die vorliegende Arbeit ist unter der finanziellen Unterstützung des Bereichs Optimierung des Fraunhofer-Instituts für Techno- und Wirtschaftsmathematik ITWM, Kaiserslautern, entstanden.
Neben der finanziellen Unterstützung, wäre diese Arbeit in diesem Umfang wohl nicht ohne die anregenden Gespräche mit Kollegen und meinen Betreuern, aber auch nicht ohne den Rückhalt meiner Freunde und Familie entstanden.

Großer Dank geht daher an meine Betreuer Prof. Dr. Karl-Heinz Küfer und PD Dr. René Henrion für die Möglichkeit, an einem sehr vielseitigen Thema arbeiten zu dürfen, das auf der theoretischen Seite viele mathematische Aspekte berührt und zeitgleich auf der Anwendungsseite Einblicke in das Wesen von Problemen zulässt. Zugleich möchte ich mich für die Einführung in das Themengebiet, die wegweisendene Gespräche und den immer wieder neu gesetzten Fokus bei beiden bedanken.
In diesem Sinne möchte ich mich auch herzlich bei Dr. Jan Schwientek, Dr. Holger Heitsch und Dr. Tobias Seidel bedanken, die mir bei fachlichen Diskussionen immer mit Rat und Tat zur Seite standen.

Für das Korrekturlesen bedanke ich mich zudem bei Dr. Dimitri Nowak, Dr. Helene Krieg, Dr. Till Heller, Tobias Joosten, Jan Schröder, David Mogalle und Lennart Künz. Ihre Anmerkungen und Korrekturen haben die Arbeit deutlich lesbarer gemacht.
Auch meinen übrigen Kollegen im Bereich Optimierung des Fraunhofer-Instituts für Techno- und Wirtschaftsmathematik ITWM danke ich sowohl für die angenehme Arbeitsatmosphäre, als auch für die Erfahrungen und Tipps zur Promotion.

Zuletzt gilt mein Dank meiner Familie und meinen Freunden, die mich während meiner kompletten Promotionszeit immer wieder aufgebaut, motiviert und teilweise auf den nötigen Abstand zur Arbeit gebracht haben. Ohne diesen Ausgleich wären die vergangenen Jahre wohl deutlich eintöniger gewesen. An dieser Stelle gilt mein besonderer Dank vor allem meiner Freundin Michelle Usslar, sowie Martin Schiffer und Kurt Schardt, ferner all denen, die zum Gelingen dieser Arbeit beigetragen haben und bisher nicht namentlich genannt wurden oder werden wollten.

Zusammenfassung

Die Optimierung unter Unsicherheit ist ein Bereich der Mathematik, der stark von Problemen der realen Welt inspiriert ist. Für den Umgang mit Unsicherheiten, z.B. bei Packungsproblemen, Bestandsmanagement oder der Koordination von Lieferketten, sind verschiedene Konzepte zur Modellierung von Unsicherheiten entstanden. Um solche Probleme mit numerischen Methoden zu lösen, müssen wir uns in der Regel zwischen Unsicherheitskonzepten mit und ohne Verteilungsinformationen entscheiden. Während Unsicherheitskonzepte ohne Verteilungsinformationen sich mehr auf einzelne Szenarien wie etwa ein worst-case Szenario konzentrieren, erlauben Unsicherheitskonzepte mit Verteilungsinformationen beispielsweise einer Menge an unsicheren Parametern, ihre Wahrscheinlichkeiten zuzuweisen.

Modelle, die komplexe Unsicherheiten darstellen, erlauben dem Modellierenden das reale Problem exakter abzubilden, erschweren aber zugleich die mathematische Handhabung.

In dieser Arbeit konzentrieren wir uns auf die numerische Behandlung von probabilistisch-robusten (probusten) Optimierungsproblemen. Obwohl einige spezielle Instanzen dieser Probleme durch Reduktion auf bereits bekannte Optimierungsprobleme gelöst wurden, haben wir keine Lösungsverfahren gefunden, die direkt auf probuste Optimierungsprobleme anwendbar sind. Die vorliegende Arbeit soll diese Lücke füllen.

Wir beginnen mit einer Verallgemeinerung des bisher bekannten Konzepts der probusten Optimierungsprobleme, um uns die Freiheit in der Modellierung zu geben, die wir für unsere Anwendungen brauchen werden. Falls geeignete Transformationen gegeben sind, können wir zeigen, dass diese verallgemeinerten probusten Optimierungsprobleme auf die gleiche Weise behandelt werden können wie die bereits bekannte Standard-Problemklasse.

Danach konzentrieren wir uns darauf, die Standard-Problemklasse mit Methoden zu lösen, die von Konzepten aus der semi-infiniten Optimierung inspiriert sind.

Zum einen bestimmen wir untere Schranken für den Optimalwert mit Hilfe einer Folge von wahrscheinlichkeitsbedingten Optimierungsproblemen. Wir konstruieren diese Probleme über Diskretisierungsschemata, die eine spezielle Bedingung erfüllen, was zur Konvergenz der dazugehörigen Schranken gegen den Optimalwert des probusten Optimierungsproblems führt. Wir zeigen u.a., dass eine feiner werdende, gleichmäßige Diskretisierung und eine angepasste Variante der Diskretisierung von Blankenship und Falk diese Bedingung erfüllen.

Zum anderen berechnen wir obere Schranken für den Optimalwert, indem wir einer Folge von Mengenapproximationsproblemen lösen. Hier ersetzen wir die Menge der zu einer Entscheidung gehörenden zulässigen Realisierungen durch eine spezielle Menge aus einer parametrisierten Mengenfamilie. Wir geben eine hinreichende Bedingung an, die garantiert, dass diese Folge von oberen Schranken gegen eine Lösung des standard probusten Optimierungsproblems konvergiert. Zusätzlich geben wir an, wie die parametrisierte Mengenfamilie unter Berücksichtigung bestimmter Strukturen in der Definition des standard probusten Optimierungsproblems gewählt werden kann.

Schließlich führen wir die Verfahren zur Berechnung der oberen und unteren Schranken zu Sandwiching-Algorithmen zusammen.

Um die verschiedenen Lösungsverfahren genauer zu verstehen, betrachten wir geometrische Packungsprobleme, die analytisch gelöst und deren Lösung visuell interpretiert werden können. Wir definieren Diskretisierungsschemata, die die Struktur der Mengen der zulässigen Realisierungen dieser Probleme nutzen. Dabei verstehen wir, dass modifizierte Diskretisierungsverfahren und eine schnelle Wahrscheinlichkeitsauswertung entscheidend dafür sind, probuste Optimierungsprobleme effizient zu lösen.
Mit diesem Verständnis lösen wir eine probuste Formulierung eines speziellen Bestandsmanagementproblems - ein Wasserreservoirproblem. Da für diese Probleme weder die Struktur der Menge der zulässigen Realisierungen, noch eine effiziente Diskretisierungsmethode bekannt ist, müssen wir diese Informationen aus der Probleminstanz gewinnen. Deswegen definieren wir eine Diskretisierungsmethode, die die Menge der Unsicherheiten nach „wichtigen" Diskretisierungspunkten absucht. Mit dieser Methode sind wir in der Lage, das probuste Wasserreservoirproblem in (annährend) zu lösen und diese Lösung mit den Lösungen anderer Unsicherheitsmodelle zu vergleichen.
Letztlich behandeln wir eine Anwendung, bei der wir qualitativ hochwertige Produkte garantieren möchten, obwohl sowohl die Qualität der gelieferten Waren, als auch der Produktionsprozess unsicher sind.

Abstract

Optimization under uncertainty is one field of mathematics that is strongly inspired by real world problems. To handle uncertainties, e.g., in packing problems, inventory management or supply chains coordination several concepts of how to model uncertainty have arisen. To solve such problems by numeric methods, we typically have to decide between uncertainty concepts with or without distributional information. While uncertainty concepts without distribution information focus more on single scenarios such as a worst-case scenario, uncertainty concepts with distribution information allow, for example, to assign a probability to a set of uncertain parameters.

As models of complex uncertainty allow the modeler to describe the problem in more details, the mathematical handling of these models gets harder.

In this thesis, we concentrate on the numerical treatment of probabilistic-robust (probust) optimization problems. Although some special instances of these problems have been dealt with by reducing them to already known optimization problems, we are not aware of any results in the literature concerning solving techniques applicable to the class of probust optimization problems themselves. This thesis aims at filling this gap.

We start by generalizing the concept of probust optimization problems known so far to be able to model our applications. Given appropriate transformations, we can show that generalized probust optimization problems can be handled the same way as the already known standard problem class.

Then we focus on solving these standard probust optimization problems using methods that are inspired by concepts from semi-infinite optimization.

On the one hand, we calculate lower bounds of the optimal value by a sequence of joint chance constrained optimization problems. We coonstruct these problems by discretization schemes satisfying a special condition which leads to the convergence of the corresponding lower bounds to the solution of the standard probust optimization problem. Furthermore, we show that, e.g., a uniform discretization approach and an adapted variant of the Blankenship and Falk discretization fulfill this condition.

On the other hand, we create upper bounds of the optimal value using a sequence of set-approximation problems. Here we substitute the set of feasible realizations that is connected to a fixed decision by a special set out of a given family of parametrized sets. We provide sufficient conditions to guarantee that the sequence of upper bounds converges towards the optimal value of the standard probust optimization problem. Additionally, we comment on how to select the family of parametrized sets based on structures within these optimization problems.

In the end, we combine the introduced upper and lower bounds to define sandwiching algorithms.

To understand the different solving methods in more detail, we consider geometric packing problems which can be solved analytically and can be interpreted visually. We define discretization schemes that use the structure of sets of feasible realizations to solve these problems. Hereby, we understand that modified discretization methods and a fast probability evaluation are critical to solve probust optimization problems efficiently.

With this understanding, we solve a probust formulation of a specific inventory management problem - a water reservoir problem. As we do not recognize a special structure in the set of feasible realizations of these problems, nor an efficient discretization method, we have to derive these pieces of information from the problem instance. Therefore, we define a discretization method that scans the uncertainty set for “important” discretization points. With this method we are able to solve the probust water reservoir problem (approximately) and compare its solution with the solutions of other uncertainty models. Ultimately, we consider an application where we want to guarantee high quality products despite the quality of delivered goods and the production process itself is uncertain.

Contents

List of symbols and abbreviations

Analysis

$\mathbb{N}$	natural numbers {1,2,3,...}
$\mathbb{Q}$	rational numbers
$\mathbb{R}$	real numbers
$\overline{\mathbb{R}}$	extended real numbers $\mathbb{R} \cup \{\pm\infty\}$
$\mathbb{R}^n$	n-dimensional Euclidean space
$\|\|x\|\|_2$	Euclidean norm, $\|\|x\|\|_2 := \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$
$\langle x, y \rangle$	Euclidean scalar product, $\langle x, y \rangle := x_1 y_1 + x_2 y_2 + ... + x_n y_n$
$B_\epsilon(x)$	closed ball around x with radius $\epsilon > 0$
$\mathbb{S}^{m-1} = \{x \in \mathbb{R}^m \mid \|\|x\|\|_2 = 1\}$	unit sphere in $\mathbb{R}^m$
$\chi_A : X \to \{0, 1\}$	indicator function of a set A with $\chi_A(x) = 1$ if and only if $x \in A$
$Df(x)$	derivative/Jacobi matrix
$D^2 f(x)$	second derivative/Hessian matrix
$D_x f(x, y)$	derivative with respect to the variable x

Topology

$(X, \mathcal{T}_X)$	set X with corresponding topology $\mathcal{T}_X$
(X, d_X)	set X with corresponding metric $d_X : X \times X \to \mathbb{R}_{\geq 0}$
$\mathrm{cl}\,(A)$	closure of a topological set A
$\mathrm{int}\,(A)$	interior of a topological set A
∂A	boundary of a topological set A
2^A	power set of a set A
$\mathcal{K}(X)$	set of compact sets within topology $(X, \mathcal{T}_X)$
$\mathcal{U}(x)$	set of neighborhoods of $x \in X$
$d_H(A, B)$	Hausdorff-distance between two compact sets A, B

Correspondences

$\Gamma : X \rightrightarrows Y$	set-valued mapping between X and Y
$f(\cdot, \Gamma) : X \to \mathbb{R}$	supremum of $f(x, y)$ over $y \in \Gamma(x)$

Semi-infinite optimization

$Y : X \rightrightarrows \mathbb{R}^q$	scenario set correspondence
GSIP	generalized semi-infinite optimization problem
Z	reference set of scenarios
$\mathcal{T}_Z : X \times Z \to \mathbb{R}^q$	transformation function for scenarios
SIP	standard semi-infinite optimization problem

Stochastic

$(\Xi, \mathcal{A})$	measurable space with σ-algebra $\mathcal{A}$
$\mathcal{P}(\Xi, \mathcal{A})$	set of probability measures on $(\Xi, \mathcal{A})$
$\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$	decision-dependent probability measure on $(\Xi, \mathcal{A})$
$(\Xi, \mathcal{A}, \mathbb{P})$	probability space with probability measure $\mathbb{P}$
$\mathbb{P}(A)$	probability of a measurable set $A \in \mathcal{A}$
$\operatorname{supp}(\mathbb{P})$	support of a probability distribution $\mathbb{P}$
$\mathbb{P}(f(\xi) \leq 0)$	probability of the set $A := \{\xi \in \Xi \mid f(\xi) \leq 0\}$
$d_D(\mathbb{P}_1, \mathbb{P}_2)$	discrepancy distance between two probability measures $\mathbb{P}_1, \mathbb{P}_2$ on $(\Xi, \mathcal{A})$
$d_{\mathbb{P}}(A, B)$	probability of the symmetric difference of two measurable sets $A, B \in \mathcal{A}$ w.r.t. $\mathbb{P}$
$\mathcal{B}(X)$	Borel-σ-algebra on the topological space $(X, \mathcal{T}_X)$
$Z : \Xi \to \mathbb{R}^m$	random vector
$\mu(Z)$	expected value of a random vector Z
$\sigma^2(Z), \Sigma(Z)$	(one or multi-dimensional) variance of a random vector Z
$\mathcal{U}(A)$	uniform probability distribution over the measurable set A
$\mathcal{N}(\mu, \Sigma)$	(multivariate) normal distribution with expected value μ and variance Σ
$F : \mathbb{R} \to [0, 1]$	cumulative distribution function of a standard normal distributed random variable
$F^{-1} : [0, 1] \to \mathbb{R}$	inverse cumulative distribution function of a standard normal distributed random variable
(J)CC	(joint) chance constrained optimization problem

Probust optimization

$f : X \to \mathbb{R}$	objective function
$p \in [0, 1]$	probability threshold
$\mathcal{F}_p$	feasible set given a threshold $p \in [0, 1]$
$\Omega(x)$	set of feasible realizations of a decision x
$\varphi : X \to [0, 1]$	outer/ probability evaluation function
$g : X \times \Xi \times T \to \mathbb{R}$	inner function
$T : X \times \Xi \rightrightarrows \mathbb{R}^q$	uncertainty set correspondence
$T_{\max} = \bigcup_{\substack{x \in X, \\ \xi \in \Xi}} T(x, \xi)$	union of all possible scenarios defined by T
GPP	generalized probust optimization problem
$(\hat{\Xi}, \hat{\mathcal{A}}, \hat{\mathbb{P}})$	reference probability space
$\mathcal{T}_{\hat{\Xi}} : X \times \hat{\Xi} \to \Xi$	transformation function for realizations
$\hat{T}$	reference set of scenarios
$\mathcal{T}_{\hat{T}} : X \times \hat{\Xi} \times \hat{T} \to \mathbb{R}^q$	transformation function for scenarios
SPP	standard probust optimization problem

Probust discretization schemes

$\varphi(\cdot, S) : X \to [0,1]$	probust function given $S \subseteq T$
$\Omega(x, S)$	set of feasible realizations of $x \in X$ given $S \subseteq T$
Φ	(sequence of) subset scheme(s)
SPP_S	approximated standard probust optimization problem with $S \subseteq T$
$\mathcal{F}_{S,p}$	approximated feasible set given a threshold $p \in [0,1]$ and a subset $S \subseteq T$

Set-approximation schemes

Δ	design space
$\delta \in \Delta$	design parameter
$D : X \times \Delta \to \mathcal{A}$	design function
$D(x, \delta)$	design
$\mathcal{D} \subseteq \mathcal{A}$	family of designs
Ψ	(sequence of) design scheme(s)
ISA_D	inner set-approximation problem using design function D
$\mathcal{F}_{D,p}$	feasible set of ISA using design function D and threshold $p \in [0,1]$
$\tilde{\mathcal{F}}_{D,p}$	projection of feasible set $\mathcal{F}_{D,p}$ onto x-component

Stochastic design-centering problems

$D : X \times \Xi \to \mathcal{A}$	design function
$C : X \times \Xi \to \mathcal{A}$	container function
SDC	stochastic design-centering problem

Water reservoir problem

T	end time point
$\omega \in \mathbb{R}^m$	inflow frequencies
$c : [0,T] \to \mathbb{R}$	cost function
$\tilde{c} \in \mathbb{R}^n$	accumulated cost vector
$\bar{l}$	upper critical filling height
$\underline{l}$	lower critical filling height
l_0	reservoir filling height at $t = 0$
$l : X \times \Xi \times T \to \mathbb{R}$	filling height function
PWR	probust water reservoir problem

Distillation

$K \in \mathbb{N}$	number of substance components [without unit]
$\dot{F} \in \mathbb{R}_{\geq 0}$	feed stream of the process in [mol/s]
$x_F \in [0,1]^K$	rate of components within the feed in [mol/mol]
$\dot{L} \in \mathbb{R}_{\geq 0}$	bottom stream in [mol/s]
$x \in [0,1]^K$	rate of components within the bottom stream in [mol/mol]
$\dot{V} \in \mathbb{R}_{\geq 0}$	head stream in [mol/s]
$y \in [0,1]^K$	rate of components within the head stream in [mol/mol]
$p \in \mathbb{R}$	partial pressure on the vaporous head stream [Pa]
$T \in \mathbb{R}$	temperature of head (and bottom) stream [K]
$p^S(T) \in \mathbb{R}^K$	temperature depending saturation vapor pressure of the different components in [Pa]
$\gamma(x,T) \in \mathbb{R}^{K \times K}$	share and temperature depending activity coefficient between two components of the substance [without unit]
$\dot{Q} \in \mathbb{R}$	heat output in [W]
$T_F \in \mathbb{R}$	feed temperature in [K]
$h^l(x,T) \in \mathbb{R}$	share and temperature depending entropy of liquid stream in [J/(mol s)]
$h^v(x,T) \in \mathbb{R}$	share and temperature depending entropy of vapor stream in [J/(mol s)]

Introduction

When real world problems are to be solved with the help of mathematical approaches, the actual work process does not begin with computing a solution of a given mathematical problem, but already one step before in the modeling. Considering a specific model, two questions quickly arise:

- Is the model realistic enough?
- Is the model (efficiently) solvable?

The first question aims for a model which is as detailed as possible. In contrast, the second question requires a manageable and thus, in a certain sense, simple model. Often these requirements are conflicting and force the modeler to make a trade-off.
One way to make this trade-off accessible is working with model parameters. Usually there are parameters that have to be set to specify the model but their value is not known to the modeler. If we use fixed values in the model, the corresponding model might lead to unrealistic results. If we allow the parameters to be fixed later, we have to handle so called parametrized models.
Such a parametrized model of a non-linear optimization problem can be denoted as

$$\mathrm{NLP}_u : \min_{x \in X} f(x,u) \text{ s.t. } g(x,u) \leq 0.$$

The objective function and the constraint depend on the parameter u. Thus, a decision x that leads to a low objective value for a specific parameter might lead to a high objective value or violate the constraint for another parameter assignment.
If we have to decide before we know how the parameter $u \in U$ is realized, u is called uncertain parameter. Our aim is to fix some decision $x^* \in X$ independent from the uncertain parameter, but in a way that it handles the possible parameter assignments "robustly". This means that the decision fulfills the constraint while ensuring a low objective value for a lot of parameter values.
Many different concepts on how to deal with uncertain parameters have been introduced in the literature.
The approaches differ on the level of information that is available for the uncertain parameters. One typically distinguishes between *uncertain parameters without* and *with distributional information.* While the first type of parameters just gives us the set of possible parameter values, the second type does also tell us how probable it is that a certain parameter attains a specific value.

Consequently, considering the first type of parameters we have to define robustness concepts based on the possible values of the uncertain parameter.
Some popular uncertainty approaches are:

- The *worst-case approach*, where we consider the parameter assignment which makes it as hard as possible for the current decision to fulfill the constraint or which has an objective value that is as high as possible. This concept was introduced by Soyster in [76] and is handled extensively in works of Ben-Tal et al. (see [11, 12]).

- The *minimum-regret approach*, where we minimize the maximal possible difference between the objective value and the objective value that we could have reached, if we would have known the uncertain parameter beforehand. This approach was introduced by Savage in [69] and is analyzed in optimization problems, e.g., by Inuiguchi et al. in [48].

- The *adjustable-robust approach*, where we can separate the decision into two parts. The first part has to be fixed before the uncertainty is known and is called here-and-now decision. The second part can be referred to as wait-and-see decision and can be fixed after the uncertain parameter is revealed. This approach was introduced by Ben-Tal et al. in [13]. Current results are presented in the survey of Yanikoglu et al. in [88].

More such concepts can be found, for example in [34].
Using the second type of uncertain parameters with distributional information, we can weight single outcomes of the uncertain parameter by the additional information. Some popular approaches are:

- *Chance constrained optimization*, where we fulfill conditions influenced by a random parameter within a given percentage of cases. This approach was introduced by Charnes et al. in [23] and the most important results can be found in works of Prékopa [61] (Chapter $8, 10, 11$) and of Shapiro et al. [75] (Chapter $1, 4$).

- *Risk concepts*, where we consider the deviation of the uncertain parameters from the expected realization and weight them based on the given distributional information. One of the earliest papers about risk measures is Rothschild et al. [67], more properties of risk measures can be found in Shapiro et al. [75] (Chapter 6).

- *Stochastic optimization with recourse* which can be interpreted as the stochastic version of an adaptive-robust approach, where the distributional information is used to evaluate the objective function. This approach was introduced by Dantzig [24] and Beale [9]. Main results for this approach can be found in Prékopa [61] (Chapter $9, 12$ and 13) and in Shapiro et al. [75] (Chapter $1, 2$ and 3).

In this thesis, we consider an uncertainty approach in which we have distributional information on part of the uncertain parameter, but not about all of it.

This implies that we can divide the uncertain parameter into the two parts

$$\begin{aligned} &u = (\xi, t) \text{ and} \\ &U = \Xi \times T. \end{aligned}$$

We assume that we have distributional information on $\xi \in \Xi$, the so called *realizations*, while we do not have such information for $t \in T$, the so called *scenarios*.
If in each constraint and the objective function, either realizations ξ or scenarios t do occur, we can use the already introduced approaches for uncertain parameters with or without distributional information individually. However, if both types of uncertain parameters appear in an expression at the same time, then this is no longer possible. Furthermore, it may happen that this combination of uncertain parameters implies new problem structures. In this thesis, we handle the scenarios $t \in T$ by a worst-case approach and the realizations $\xi \in \Xi$ by a chance constrained approach which leads to two possible models. To be able to distinguish these models, it is useful to introduce the *set of feasible realizations*

$$\Omega(x, t) = \{\xi \in \Xi \mid g(x, \xi, t) \leq 0\}.$$

This set represents all realizations that fulfill the uncertain constraint for a decision $x \in X$ and a scenario $t \in T$.
We can then either focus on the worst-case probability over these single sets resulting in the *robust-probabilitstic (robubilistic) model*

$$\text{RP}: \quad \min_{x \in X} f(x) \text{ s.t. } \mathbb{P}(g(x, \xi, t) \leq 0) \geq p \ \forall t \in T,$$

or we consider the probability of the intersection of all these sets representable by the *probabilistic-robust (probust) model*

$$\text{SPP}: \quad \min_{x \in X} f(x) \text{ s.t. } \mathbb{P}(g(x, \xi, t) \leq 0 \ \forall t \in T) \geq p.$$

Because the intersection of all sets is in general smaller than each single set, the probust model is more restrictive than the robubilistic one.
Although both problems RP and SPP might resemble distributionally robust optimization problems that deal with parameter-depending probability distributions (see e.g., [62, 70]), they are not such. In contrast, probust and robubilistic models handle parameter-dependent events which are evaluated by a constant probability measure.
As probust optimization problems are our main point of mathematical interest, we briefly comment on how they are used in literature until now.
So far probust optimization problems are considered in papers by Grandón et al. [37] and Adelhütte et al. [5] to model problems connected to gas networks. The solving procedures in these papers are based on reducing the probust optimization problem to a chance constrained optimization problem. This is achieved by finding an analytical expression of the worst-case scenario for all possible decisions and realizations of the uncertain parameter with distributional information.
Adelhütte et al. have mentioned in [5] while analyzing gas networks with circles that such a representation can be hard to find and is not given in general.
To the best of our knowledge there are no algorithms available to solve probust optimization problems if no analytic description of the worst-case scenarios is known. In this thesis, we aim for introducing such algorithms.

Scope of this work

The problem description of probust optimization problems requires that a constraint is fulfilled for all scenarios $t \in T$. This reminds us of semi-infinite optimization problems

$$\text{SIP} : \min_{x \in X} f(x) \text{ s.t. } g(x,t) \leq 0 \ \forall t \in T$$

which also deal with this structure.
There are several methods to solve semi-infinite optimization problems (see [63, 80] or Chapter 5 of [78]). Since numerical solution approaches often work on some discretized basis, it seems convenient to investigate discretization schemes from semi-infinite optimization and adapt them to the probust context. One difficulty in this adaption is that, in general, we cannot find one single scenario that represents the worst-case for a fixed decision as in the semi-infinite case.
In the probust context, we have to handle a semi-infinite constraint for each realization which leads to a family of scenarios $(t(\overline{x}, \xi))_{\xi \in \Xi}$ that represent the worst-case given a decision $\overline{x} \in X$ and the realization $\xi \in \Xi$ via

$$\max_{t \in T} g(\overline{x}, \xi, t) = g(\overline{x}, \xi, t(\overline{x}, \xi)).$$

As we are mainly interested in continuous distributions later on, this requires to calculate an infinite number of worst-case scenarios for a fixed decision $\overline{x}$.
Although we cannot neglect realizations in general, we can ask if we need the whole uncertainty set of scenarios or just a (finite) subset $S \subseteq T$.
This question leads to the definition of the *candidate-condition*

$$\forall \epsilon > 0, t \in T : \mathbb{P}(g(\overline{x}, \xi, s) \leq 0 \ \forall s \in S) - \mathbb{P}(g(\overline{x}, \xi, s) \leq 0 \ \forall s \in S \cup \{t\}) < \epsilon$$

which checks whether the addition of a scenario to the considered subset affects the overall probability or not. If not, the given subset is a suitable substitute for the whole set of scenarios. Working with a sequence of discretizations $(T_k)_{k \in \mathbb{N}}$ that induces a sequence of joint chance constrained subproblems

$$\text{SPP}_{T_k} : \min_{x \in X} f(x) \text{ s.t. } \mathbb{P}(g(x, \xi, t) \leq 0 \ \forall t \in T_k) \geq p,$$

we can check if the corresponding solutions converge towards a solution of the original probust optimization problem.
In this work, we show that some discretization schemes from semi-infinite optimization also converge in the probust setting, e.g., uniform discretization schemes or modified versions of the adaptive discretization approach from Blankenship and Falk introduced in [17].
Because the constraints indexed by $t \in T \backslash T_k$ are not considered in these subproblems, the iterates are in general infeasible w.r.t. the original probust optimization problem.
As we do not know to which degree our current iterate is infeasible, we have to formulate an appropriate stopping criterion for iterative discretization schemes.
To be able to find such a stopping criterion, we reconsider the probust optimization problem to develop a second solution approach to generate feasible iterates.

So far, we interpreted probust optimization problems as semi-infinite optimization problems depending on some random value $\xi \in \Xi$. This interpretation focuses on the scenarios, while the realization just appears as an index.
Readjusting our perspective, we focus on the parameter combination $u = (\xi, t)$. Because the probust constraint can also be noted as a set-wise condition, we are interested in how to select an appropriate subset of U that fulfills this condition. While the worst-case approach causes to respect all scenarios, we are allowed to ignore some realizations in the set of realizations.
As the search space of subsets is quite big, we reduce this search space to a set $\Delta \subseteq \mathbb{R}^d$ by parametrizing "interesting" subsets $D \subseteq \Xi$ by *design variables* $\delta \in \Delta$ to formulate the *inner set-approximation problem* for probust optimization problems

$$\mathrm{ISA}_D : \min_{x\in X, \delta\in\Delta} f(x) \text{ s.t. } \mathbb{P}(D(x,\delta)) \geq p,$$
$$g(x,\xi,t) \leq 0 \ \forall \xi \in D(x,\delta), t \in T.$$

This problem can be interpreted as a (generalized) semi-infinite optimization problem with an additional probability evaluation constraint.
As we reduced the search space from the (measurable) subsets of Ξ to the set Δ, a solution of the inner set-approximation problem is feasible w.r.t. the original probust optimization problem.
First, we comment on how to define "interesting" subsets depending on the structure of the constraint g. Then, we give a sufficient condition such that an iterative set-approximation scheme converges to the optimal solution of the corresponding probust optimization problem.
With these set-approximation schemes, we can calculate feasible iterates and upper bounds for the optimal objective value of a probust optimization problem. Together with the lower bounds defined by a discretization scheme, we define a sandwiching procedure with a well-defined stopping criterion.
We can even use the information calculated from discretization schemes to define specialized set-approximations such as

$$D \equiv \bigcap_{t\in T_k} \{\xi \in \Xi \mid g(x_k, \xi, t) \leq 0\},$$

where (x_k, T_k) denotes the output of the k-th iteration of the discretization method. Although we do not give a convergence guarantee for this special definition of D, it performs well in most of our applications. This is because the corresponding inner set-approximation is a standard semi-infinite optimization problem which can be solved efficiently, e.g., by the adaptive discretization approach by Blankenship and Falk.

To test and compare the introduced numerical schemes for solving probust optimization problems, we consider probust optimization problems that we can also solve analytically. Motivated by design-centering problems (see ,e.g., [39]) and their interpretation as semi-infinite optimization problems, we slightly modify the problem definition to generate probust optimization problems by assuming that the designs or container are influenced

by a random disturbance. As these problems can be solved analytically and can be represented geometrically, we can compare and visualize the numerical results of our probust solution schemes.
With this experience we solve probust optimization problems which are strongly inspired by applications of water reservoir management and chemical process engineering.

Structure of this thesis

This thesis is structured as follows. It consist of two parts, where the first part consisting of Chapter 1 to 4 concentrates on introducing algorithms to solve probust optimization problems. The second part including Chapter 5 to 7 highlights the numerical behavior of the introduced solution schemes and show how we can make the theoretical results usable for realistic applications.

In **Chapter 1**, we introduce a generalized version of probust optimization problems as they are known in literature. We present basic problem classes like chance constrained optimization problems and semi-infinite optimization problems in Section 1.1. After surveying how we can generalize these concepts using decision-dependent uncertainties in Section 1.2, we introduce generalized probust optimization problems in Section 1.3.

Chapter 2, then, introduces the first numerical solution method for probust optimization problems in form of probust subset schemes. To understand which discretization points are important, we focus on simpler problems in Section 2.1. After that, we define the probust discretization algorithm in Section 2.2 and give sufficient conditions for its convergence. We, then, apply the convergence theorem for some example discretization schemes that are inspired by schemes from semi-infinite optimization in Section 2.3.

The main idea of **Chapter 3** is to generate feasible iterates by utilizing set-approximation schemes. We start with approximating the probability of some (measurable) set by shrinking down the search space from the corresponding σ-algebra to a family of reference sets in Section 3.1.
We use these insights to approximate solutions of chance constrained optimization problems by solutions of inner set-approximation problems in Section 3.2. After defining an iterative inner set-approximation approach in Section 3.3, we discuss an example of how to use it with a probust optimization problem in Section 3.4.

To combine the introduced solution schemes in **Chapter 4**, we discuss how we can use information from probust subset schemes in set-approximation schemes and vice versa. We consider some examples that illustrate how probust subset schemes and set-approximation schemes have some kind of antagonistic convergence assumptions. Therefore, we cannot expect convergence of the resulting sandwiching schemes in general. Nevertheless, we find an interesting sandwiching scheme that we use to create bounds for the objective values in the application part.

To be able to discuss the performance of different probust solution schemes, we introduce a class of analytically and visually controllable problems in **Chapter 5**. Since these problems are described by several inequality constraints, we introduce solution approaches which are able to handle these constraints separately.

Motivated by the results of Chapter 5, we define a specialized probust solution scheme to solve water reservoir problems in **Chapter 6**. It combines an adaptive discretization approach with an uniform discretization. We then compare the results of the probust model with solutions of a robubilistic and an expected value model.

In **Chapter 7** we consider a flash distillation problem from chemical process engineering. The corresponding model states equality constraints which have to be handled in the probust setting. We develop a solution approach based on the implicit function theorem and numerical tests to solve this problem instance. Afterwards we discuss the results of the probust formulation for different parameters of the underlying uncertainty sets.

We end this thesis with a summary and a suggestion of future work.

Part I

Theory

1 Probust optimization

In this chapter, we introduce the basics of probust optimization. Our goal is to introduce basic concepts for the analyses following up in the first part of this thesis and to create a modelling framework that allows us to handle the applications in Part II.

We start with standard probust optimization problems in Section 1.1, where we refer to closely related problem classes as well as to corresponding analytical statements and numerical tools. These problem classes will reappear later in Chapter 2 and Chapter 3, where we consider iterative solution approaches to handle probust optimization problems. After defining the standard probust optimization problem and surveying theoretical results that are known in literature so far, we give an example of how to solve such a standard probust optimization problem analytically. We also visualize single steps of the solving procedure that will be referred to later in Part II of this thesis.

As the applications in Part II cannot be modelled by standard probust optimization problems directly, we recall decision-dependent uncertainty concepts in Section 1.2 and use them to introduce generalized probust optimization problems in Section 1.3. These extended problem class gives us the freedom to model all applications considered in the second part of this thesis.

Although the modelling perspective needs an extended concept of probust optimization to formalize the problems in Chapter 5, the problem structure of these problems stays the same if we can find appropriate transformations of the uncertainty set. This reduction of generalized probust optimization problems to standard probust optimization problems is the main result of this chapter regarding the analytical perspective.

1.1 Standard probust optimization problems

To achieve a better understanding of the solution existence results which are presented in literature for standard probust optimization problems and to prepare the theoretical background for analyses in Chapter 2 and Chapter 3, we introduce concepts from semi-infinite and chance constrained optimization. These problem classes are connected quite naturally with the probust setting by fixing either a realization $\xi \in \Xi$ to get semi-infinite optimization problem or by fixing a scenario $t \in T$ to get a chance constrained optimization problem.

One of the first steps to handle an optimization problem is to talk about a well-defined solution of the problem. Therefore, we start with such results from semi-infinite optimization.

Semi-infinite optimization

(Standard) semi-infinite optimization problems have the form

$$\mathrm{SIP}: \min_{x \in X} f(x) \text{ s.t. } g(x,t) \leq 0 \;\forall t \in T,$$

where $X \subseteq \mathbb{R}^n$ is some set, $T \subseteq \mathbb{R}^q, q \in \mathbb{N}$ is the set of scenarios of infinite cardinality and $f: X \to \mathbb{R},\; g: X \times T \to \mathbb{R}$ are functions. Because the decision variable is finite dimensional while we handle infinite many constraints this leads to the name *semi-infinite*.
If we compare SIP to the worst-case approach introduced by Ben-Tal in [12], we use basic analysis to see that semi-infinite optimization problems can be rewritten as robust optimization problems with an infinite number of scenarios $t \in T$.
The existence of a solution of a SIP can be shown by

Lemma 1.1.1 (Continuous constraint in SIP, Lem. 16.29 in [6]))
Let $X, T \neq \emptyset$ be two topological spaces, $g: X \times T \to \mathbb{R}$ be a lower semi-continuous function w.r.t. $(x,t) \in X \times T$. Then $\varphi: X \to \mathbb{R}, x \mapsto \sup_{t \in T} g(x,t)$ is lower semi-continuous.

We can use this lemma with a compact set X and a lower semi-continuous function $f: X \to \mathbb{R}$ to guarantee that the induced SIP has a well-defined solution $f^* \in \mathbb{R}$ and a well-defined minimizer $x^* \in X$ if the corresponding feasible set is not empty.
We can also talk about convexity and therefore about an unique solution of a SIP using the following theorem from Still:

Theorem 1.1.2 (Convex feasible set of SIP, Thm. 4*a* in [82])
Given a SIP with convex objective function $f: X \to \mathbb{R}$ and convex $g(\cdot,t): X \to \mathbb{R}$ for all $t \in \mathbb{R}$. Then the feasible set of SIP is convex.

If we additionally know that f is strictly convex and the assumptions from the last theorem are satisfied, then we know that SIP has an unique minimizer.
Next to this analytical properties of SIPs, we are also interested in how we can solve SIPs numerically. In this context the following subproblem - called *lower-level problem* - is frequently used and defined by fixing some decision $\overline{x} \in \mathbb{R}^n$

$$Q(\overline{x}) \;:\; \max_{t \in T} g(\overline{x}, t). \tag{1.1}$$

Numerical solution approaches for SIPs can be categorized according to the surveys from Reemtsen et al. [63] and Stein [80] as follows:

1. Discretization methods that reduce the set T to a finite subset. Useful discretization strategies can be found, e.g., in papers by Blankenship and Falk [17], Reemtsen [64] and Seidel [73].

2. Reformulation of the lower-level problem using Karush-Kuhn-Tacker-conditions. As these reformulations imply so called complementarity constraints, they are numerically challenging and therefore often smoothed, see e.g., Stein's book [78].

3. Local reduction methods utilizing stronger assumptions to be able to reduce the semi-infinite optimization problem locally to a (finite) nonlinear optimization problem. The corresponding theory can be found, e.g., in Klatte's paper [50].

As probust optimization is a quite young research field, we will work in this thesis with discretization methods. Not only do they allow us to handle problems with just a few assumptions which is analytically attractive, but they also define a mapping from the set T with infinite cardinality to a discrete subset $S \subseteq T$ which we need for numerical evaluations.

Furthermore, if we have additional structure given as in Seidel's paper [73], there exist promising discretization schemes with fast convergence rates.

One popular discretization scheme to solve semi-infinite optimization problems that will inspire us in Section 2.3 is the adaptive approach from Blankenship and Falk that is introduced in [17]. It can be formulated as:

Algorithm 1 Adaptive discretization algorithm from Blankenship and Falk [17]

```
Inputs:
    Semi-infinite problem instance SIP, starting decision $x_0 \in X$, starting
    discretization $T_0 \subseteq T$
Initialize:
    $k := 0$
do
    $x_{k+1} \leftarrow \arg\min_{x \in X} f(x)$ s.t. $g(x,t) \leq 0 \ \forall t \in T_k$
    $t_k^* \leftarrow \arg\max_{t \in T} g(x_{k+1}, t)$
    $T_{k+1} \leftarrow T_k \cup \{t_k^*\}$
    $k \leftarrow k + 1$
while $g(x_k, t_k^*) > 0$
Results:
    Sequence $(x_k, T_k)_{k \in \mathbb{N}}$
```

This algorithm guarantees that all accumulation points of $(x_k)_{k \in \mathbb{N}}$ are minimizers of the original SIP if f, g are continuous and X, T are compact (see Theorem 2.1 in [17]). Considering Lemma 1.1.1 these are mild assumptions.

(Joint) chance constrained optimization

Chance constrained optimization problems have the form

$$\mathrm{CC}:\ \min_{x\in X} f(x) \text{ s.t. } \mathbb{P}(g(x,\xi)\leq 0)\geq p,$$

where $X\subseteq\mathbb{R}^n$ is some set, $(\Xi,\mathcal{A},\mathbb{P})$ is a measurable space, $p\in[0,1]$ is a probability threshold, $f:X\to\mathbb{R}$ is a continuous function and $g:X\times\Xi\to\mathbb{R}$ is measurable w.r.t. $\xi\in\Xi$ for each fixed $x\in X$ as well as continuous w.r.t. $x\in X$ for $\mathbb{P}$ almost surely all $\xi\in\Xi$. Such a function g is also called *Caratheodory function* (see e.g., [51]). In comparison to the semi-infinite optimization problem, we do not have to consider all elements $\xi\in\Xi$, but we can limit ourselves to a set that covers the probability a given percentage p of the whole set. If the function g is given by a maximum of (finitely) many functions $(g_i)_{i\in I}$, we call the corresponding problem a *joint chance constrained optimization problem* that can be noted as

$$\mathrm{JCC}:\ \min_{x\in X} f(x) \text{ s.t. } \mathbb{P}(g_i(x,\xi)\leq 0\ \forall i\in I)\geq p.$$

The existence of a solution of a CC is based on statements of Raik that are written in Russian and ,e.g., referenced by Prékopa as

Theorem 1.1.3 (Continuous constraint in JCC, Thm. 10.1.1 in [61])
Given a JCC assume that the functions $g_i:X\times\Xi\to\mathbb{R}$ are point-wise defined for $(x,\xi)\in X\times\Xi$ and lower semi-continuous w.r.t. $x\in X$ for all $i\in I,\xi\in\Xi$. Then the function $\varphi:X\to[0,1],x\mapsto\mathbb{P}(g_i(x,\xi)\leq 0\ \forall i\in I)$ is upper semi-continuous.
If additionally the condition $\mathbb{P}(g_i(x,\xi)=0\ \forall i\in I)=0$ holds for all $x\in X$ and $g_i(\cdot,\xi)$ is continuous w.r.t. $x\in X$ for arbitrary fixed $\xi\in\Xi,i\in I$, then φ is continuous as well.

Consequently, we can use these assumptions together with a compact set X and a lower semi-continuous objective function $f:X\to\mathbb{R}$ to conclude the existence of a well-defined minimizer of JCC if the corresponding feasible set is not empty.
Before we can discuss the uniqueness of a solution of a JCC, we need a concavity concept for probability measures. One such concept can be found, e.g., in the book of Shapiro et al. [75]:

Definition 1.1.4 (α-concave probability distributions)
Let $(\Xi,\mathcal{B},\mathbb{P})$ be a probability space, where $\mathcal{B}$ is the Borel-σ-algebra of Ξ. We call $\mathbb{P}$ an α-concave probability distribution, if for all $A_1,A_2\in\mathcal{B},\lambda\in[0,1]$ we can guarantee

$$\mathbb{P}(A(\lambda))\geq m_\alpha(\mathbb{P}(A_1),\mathbb{P}(A_2),\lambda),$$

where $A(\lambda)=\lambda A_1+(1-\lambda)A_2$ and for fixed $\alpha\in\mathbb{R}$

$$m_\alpha:\mathbb{R}_{\geq 0}\times\mathbb{R}_{\geq 0}\times[0,1]\to\mathbb{R},$$
$$m_\alpha(a,b,\lambda)=\begin{cases} a^\lambda b^{1-\lambda} \text{ if } \alpha=0,\\ \max\{a,b\} \text{ if } \alpha=\infty,\\ \min\{a,b\} \text{ if } \alpha=-\infty,\\ (\lambda a^\alpha+(1-\lambda)b^\alpha)^{\frac{1}{\alpha}}, \text{ otherwise.}\end{cases}$$

With this concept, we can give a sufficient condition on the convexity of the feasible set of a JCC:

Corollary 1.1.5 (Convex feasible set of JCC, Cor. 4.41 in [75])
Let $X \subseteq \mathbb{R}^n$ be some convex set, $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, I some finite index set, $g_i : X \times \Xi \to \mathbb{R}$ be Caratheodory functions that are quasi-convex w.r.t. $(x, \xi) \in X \times \Xi$ for all $i \in I$ and $p \in [0, 1]$. Let $\mathbb{P}$ be an α-concave probability measure for some $\alpha \in \mathbb{R}$ and define $\varphi : X \to \mathbb{R}, x \mapsto \mathbb{P}(g_i(x, \xi) \leq 0 \; \forall i \in I)$. Then the set $\{x \in X \mid \mathbb{P}(g_i(x, \xi) \leq 0 \; \forall i \in I) \geq p\}$ is convex and closed.

Example 4.9 in the book of Shapiro et al. [75] ensures that the probability measure induced by a multivariate normal distributed random variable is 0-concave. This example is important as we consider exactly this probability measure in Part II of this thesis.
After introducing the basic results for solution existence and uniqueness of JCCs, we present numerical solution approaches to calculate these solutions. Such approaches for JCCs can be categorized into:

1. Scenario approaches which use a finite number of realizations and considers a robust optimization problem defined on this finite sample. Because these samples do not necessarily represent the whole set of possible realizations Ξ and we are analyzing a robust optimization problem, it is not intuitively clear how solutions of this approach behave. Results can be found, e.g., in papers by Calafiore et al. [21] and Luedtke et al. [56].

2. Expected value methods for which we rewrite $\mathbb{P}(g(x, \xi) \leq 0) = \mathbb{E}(\chi_{g(x,\xi) \leq 0})$ and then use methods for evaluating an expected value like a sample average approach. Such approaches can be found, e.g., in a paper of Pagnoncelli et al. [59].
As the indicator function χ_A of an arbitrary set A is discontinuous in general, the expected value operator might be hard to analyse. By smoothing it, we get more structure to use in analyses. Papers by Hu et al. [47], Shan et al. [74] and Geletu et al. [31] deal with that topic.

3. Set-approximation methods, where the set of feasible realizations corresponding to a fixed decision x is approximated. Using this set of feasible realizations we can state a robust optimization problem.
Considering linear inequality constraints to describe the set of feasible realizations, Yuan et al. study in [89] sets like hyper-boxes and ellipses for their robust approach. Moreover, Margellos et al. [57] use hyper-boxes to approximate the set of feasible realizations, where the hyper-box is defined adaptively by an associated subproblem.

We will combine set-approximation methods with concepts from semi-infinite analysis to define inner approximations of probust optimization problems in Chapter 3.
As we will focus on probability measures that have a continuous Lebesgue-density and especially on multivariate Gaussian distributions in Part II of this thesis, we use the so called *spheric-radial decomposition* (SRD) described, e.g., in Section 2 of Déak's paper [25] or in Chapter 4 by the book of Genz and Bretz [32].

The main idea of the spheric-radial decomposition is to sample from the unit-sphere $\mathbb{S}^{m-1}$ instead of sampling from $\mathbb{R}^m$ using the radial symmetry of the random vector Z. Therefore, we decompose it into a radial part $r \geq 0$ and a direction $v \in \mathbb{S}^{m-1}$. Denoting L as the Cholesky-decomposition of the covariance matrix Σ, we can reformulate the probability induced by the Gaussian distributed random variable $Z \sim \mathcal{N}(\mu, \Sigma)$ of some measurable set Ω as

$$\mathbb{P}(\Omega) = \int_{v \in \mathbb{S}^{m-1}} \mu_\chi(\{r \geq 0 \mid \mu + rLv \in \Omega\}) d\mu_\zeta(v),$$

where μ_χ is the measure of the χ-distribution with $m-1$ degrees of freedom and μ_ζ is the uniform distribution over $\mathbb{S}^{m-1}$.
E.g., this method is used in a paper of Gotzes et al. [36], where the numerical experience in Section 7 shows that the SRD yields a faster stabilization of results than generic Monte-Carlo samples.
Additionally to this numerical experience, the integral representation by SRD allows us to calculate gradients of the probability evaluation as shown in the articles by van Ackooij et al. [1, 2] for (joint) chance constraints.
The most interesting result of these papers is the representation of the gradient of the probability evaluation. As this result needs a lot of technical definitions, we just present the main idea. In chance constrained optimization, we are interested in the set of feasible realizations corresponding to some decision $x \in X$ given by

$$\Omega(x) = \{\xi \in \Xi \mid g_i(x, \xi) \leq 0 \ \forall i \in I\}.$$

Under the assumptions of Theorem 4.1 in [2] we can switch the gradient and the integral operator to get

$$\begin{aligned}\nabla \varphi(x) &= \nabla \mathbb{P}(\Omega(x)) \\ &= \int_{v \in \mathbb{S}^{m-1}} \nabla_x \mu_\chi(\{r \geq 0 \mid \mu + rLv \in \Omega(x)\}) d\mu_\zeta(v).\end{aligned}$$

As $\Omega(x)$ can be described by functions $g_i(x, \cdot), i \in I$, we can describe the change of the probability function φ by the change of the probability density and the change of the set of feasible realizations (see also Theorem 2.1 in [83]). The probability density is constant w.r.t. $x \in X$, but the change of the set of feasible realizations can be described by the movement of its boundary $\partial\Omega(x)$ which is - given continuous constraints - a subset of

$$\{\xi \in \Xi \mid \max_{i \in I} g_i(x, \xi) = 0\}.$$

If we interpret the realizations $\xi \in \Xi$ as additional constraint indices, we can interpret $\partial\Omega(x)$ as a subset of the active indices here. Because the set of active indices is described by an equality constraint, we remember the implicit function theorem to understand the gradient representation in Theorem 4.1 by the paper of van Ackooij et al. [2]

$$\nabla \varphi(x) = \int_{v \in \mathbb{S}^{m-1}} -\frac{\nabla_x g_{i(v)}(x, \mu + r(x,v)Lv)}{\langle \nabla_z g_{i(v)}(x, \mu + r(x,v)Lv), Lv \rangle} \chi(r(x,v)) d\mu_\zeta(v), \tag{1.2}$$

where μ_ζ is the uniform distribution over the unit sphere $\mathbb{S}^{m-1}$, χ is the probability density function of a one-dimensional chi-distribution with $m-1$ degrees of freedom, $r(x,v) \geq 0$ is the length of the ray starting at μ in direction $v \in \mathbb{S}^{m-1}$ intersected with $\Omega(x)$, L is the Cholesky-decomposition of the covariance matrix Σ and $i(v)$ is the active index of $\max_{i\in I} g_i(x, \mu + r(x,v)Lv)$.

We will approximate these gradients using a sample average approach

$$\nabla\varphi(x) \approx -\frac{1}{N}\sum_{k=1}^{N} \frac{\nabla_x g_{i(v_k)}(x, \mu + r(x,v_k)Lv_k)}{\langle \nabla_z g_{i(v_k)}(x, \mu + r(x,v_k)Lv_k), Lv_k\rangle}\chi(r(x,v_k)), \tag{1.3}$$

in Part II of this thesis to solve probust optimization problems numerically, where the vectors $v_k \in \mathbb{R}^m$ with $k = 1, ..., N, N \in \mathbb{N}$ is some sample of the unit sphere $\mathbb{S}^{m-1}$.

Standard probust optimization

After having described basic properties of semi-infinite and chance constrained optimization problems which can be interpreted as probust optimization problems with fixed realization $\xi \in \Xi$ or fixed scenario $t \in T$ respectively, we are ready to consider the current results referring to (standard) probust optimization problems. We remember the definition of a (standard) probust optimization problem

$$\text{SPP}: \quad \min_{x\in X} f(x) \text{ s.t. } \mathbb{P}(g(x,\xi,t) \leq 0 \ \forall t \in T) \geq p.$$

In the remainder of this thesis, we will call g the *inner function* of a probust optimization problem and $\varphi : X \to [0,1], x \mapsto \mathbb{P}(g(x,\xi,t) \leq 0 \ \forall t \in T)$ the *outer function* of a probust optimization problem.

Because the probust optimization as a research field is quite new and no survey regarding this topic has been published so far, we give a rough overview of the existing literature:

The idea to consider semi-infinite optimization problems with additional stochastic uncertainties was stated the first time in [54]. In this article a smoothing algorithm from semi-infinite optimization was applied to a kind of probust optimization problem. It has been shown that the corresponding iterative algorithm converges to a solution of the original problem without given numerical example.

Independent of that paper, in 2017 Grandón et al. [37] handled the problem of how to minimize costs of a gas manufacturer such that the gas pipe network delivers enough gas to all costumers where their demand and the state of the gas pipes are treated as uncertain parameters. While the basic model in this paper is a probust optimization problem, it can be reduced to a joint chance constrained problem by assuming that the graph structure is a tree (see Corollary 1 in [36]).

In 2018 Farshbaf-Shaker et al. investigated in [30] the continuity, convexity and stability of probust constraints w.r.t. a changing distribution function. They interpreted probust constraints as joint chance constraints indexed by an area and connected their results to partial derivative equations (PDEs). Nevertheless, they did not computed a solution in this context nor gave an solution algorithm, either.

The first time an optimization problem was called a probust optimization problem was in 2018, when Adelhütte et al. [5] studied gas network applications.
In 2019 Grandón referred in her PhD thesis [38] towards probust constraints, but just used the vast modelling scope that this problem class allows to start her analysis of a gas network problem. The most important result for our numerical treatment of probust optimization problems can be found in van Ackooij et al. [3] where (sub-)gradient formulas from the joint chance constrained context were generalized to probust constraints.
As we will need this result as well as the results from Farshbaf-Shaker et al. in Section 1.3, we present them in more detail:

Proposition 1.1.6 (Upper semi-continuity of outer function, Prop. 1 from [30]) *Assume that X is a Banach space, $T \subseteq \mathbb{R}^q$ is a compact set, $g(x,\cdot,t)$ is Borel measurable for all $x \in X, t \in T$ and that $g(\cdot,\xi,t)$ is weakly sequentially lower semi-continuous (w.s.l.s.) for all $\xi \in \Xi$ and $t \in T$. Then*

$$\varphi : X \to [0,1], x \mapsto \mathbb{P}(g(x,\xi,t) \le 0 \ \forall t \in T)$$

is weakly sequentially upper semi-continuous (w.s.u.s.).

Proposition 1.1.7 (Lower semi-continuity of outer function, Prop. 2 from [30]) *Assume that X is a Banach space, $T \subseteq \mathbb{R}^q$ is a compact set, $(\Xi, \mathcal{A}, \mathbb{P})$ is a probability space, $g : X \times \Xi \times T \to \mathbb{R}$ is w.s.u.s. as a function of all three variables simultaneously. Then φ is w.s.l.s. at all $x \in X$ satisfying*

$$\mathbb{P}\left(\sup_{t\in T} g(x,\xi,t) = 0\right) = 0.$$

Proposition 1.1.8 (Convexity of feasible set of SPP, Prop. 4 from [30]) *Let X be an arbitrary vector space and T be an arbitrary index set. Let the m-dimensional random vector ξ have a log-concave density (i.e., a density whose logarithm is a possibly extended-valued concave function). Finally, assume that $g(\cdot,\cdot,t)$ is quasi-convex for all $t \in T$. Then, the set*

$$\mathcal{F}_p := \{x \in X \mid \varphi(x) \ge p\}$$

is convex for any $p \in [0,1]$.

Based on the results for joint chance constraints, 2019 van Ackooij et al. generalized the representation of a gradient from the joint chance constrained case to the probust case in their paper [3]. Therein, in Corollary 4.5 and Corollary 4.6 they give sufficient conditions for the function

$$\varphi : X \to [0,1], x \mapsto \mathbb{P}(g(x,\xi,t) \le 0 \ \forall t \in T(x))$$

to be Lipschitzian and differentiable at a given point $\overline{x} \in X$. The main difference to the joint chance constrained case (1.2) is that we have to respect changes of the active index w.r.t. the decision as well.

Using these gradients, we could apply gradient-based optimization methods to find a solution of a probust optimization problem numerically. The main issue making such methods unattractive in practice is that we do not know the active indices in general.
In contrary, we do not need gradient formulations of probust constraints if we face a "simple" probust optimization problem. Therefore, we end this section by solving a probust optimization problem analytically and introduce a visualization of the solution process.
We solve a (standard) probust optimization problem by the following three solution steps:

1) Fix an arbitrary $(x, \xi) \in X \times \Xi$ and calculate $g(x, \xi, T) := \max_{t \in T} g(x, \xi, t)$.

2) Fix an arbitrary $x \in X$ and calculate $\varphi(x) := \mathbb{P}(g(x, \xi, T) \leq 0)$.

3) Solve $\min_{x \in X} f(x)$ s.t. $\varphi(x) \geq p$.

We go through these singe steps in the next example:

Example 1.1.9 (Solving a probust optimization problem analytically)
Let $X := [0,1], f : X \to \mathbb{R}, x \mapsto x, (\Xi, \mathcal{A}, \mathbb{P}) = ([0,1], \mathcal{B}, \mathbb{P})$, *where* $\mathcal{B}$ *is the Borel-*σ*-algebra on* $[0,1]$ *and* $\mathbb{P}$ *is the probability measure induced by* $\xi \sim \mathcal{U}([0,1])$, $p = 0.9$, $T := [-2,2]$ *and* $g : X \times \Xi \times T \to \mathbb{R}, (x, \xi, t) \mapsto -x + \xi - (x + \xi + t)^2$. *Then we can solve the probust optimization problem*

$$\min_{x \in X} f(x) \text{ s.t. } \mathbb{P}(g(x, \xi, t) \leq 0 \ \forall t \in T) \geq p$$

using the three solution steps:
1) *Fix an arbitrary* $x \in [0,1]$, *fix an arbitrary* $\xi \in [0,1]$, *then we can calculate*

$$g(x, \xi, T) = \max_{t \in [-2,2]} g(x, \xi, t) = -x + \xi.$$

2) *Fix an arbitrary* $x \in [0,1]$ *and calculate*

$$\begin{aligned} \varphi(x) &= \mathbb{P}(-x + \xi \leq 0) \\ &= \mathbb{P}(\xi \leq x) = x. \end{aligned}$$

3) *Solve* $\min_{x \in [0,1]} x$ *s.t.* $x \geq 0.9$. *This leads directly to the (unique) minimizer* $x^* = 0.9$.

Because an analytical solution is not always given for probust optimization problems or single solution steps, we are interested in how we can visualize the solving procedure.
Therefore, we go through the single calculation steps of the last example once again, but visualize the interesting subsets in the decision-realization-scenario space $X \times \Xi \times T$:
In the first step, we fix a decision $x_0 = 0.5 \in [0,1]$ and a realization $\xi_0 = 0 \in [0,1]$ to calculate a corresponding worst-case scenario that $t^*(x_0, \xi_0) \in T$ which represents the whole set $T = [-2,2]$. We have seen in Example 1.1.9 step 1) that the worst-case scenario is $t^*(x, \xi) = -x - \xi$ for any $x \in [0,1], \xi \in [0,1]$. This means we can substitute the interval T by $\{-0.5\}$ concerning the maximium operator.

Because the decision x_0 and the realization ξ_0 are chosen arbitrarily, we plot them as red crosses that might change later in the solution procedure. The worst-case scenario defined by these values and visualized as a green cross in Figure 1.1.

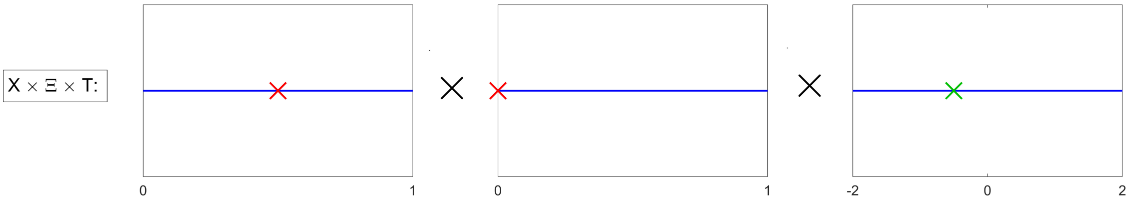


Figure 1.1: Fixed decision $x_0 \in X$ (red cross on the left), realization $\xi_0 \in \Xi$ (red cross in the middle) and corresponding worst-case scenario $t^*(x_0, \xi_0) \in T$ (green cross on the right)

Now we are entering step 2), where we have to calculate $\varphi(x_0)$ for the fixed $x_0 \in X$. To do so, we have to check which realizations $\xi \in \Xi$ satisfy $g(x_0, \xi, T) \leq 0$. In Example 1.1.9 we calculated that the corresponding set is $\Omega(x_0) = [0, 0.5]$. Please note that every realization $\xi \in \Omega(x_0)$ leads to a new worst-case scenario $t^*(x_0, \xi) \in T$. Because this set is defined by the choice of x_0, we print it with its corresponding worst-case scenarios as green line segments in Figure 1.2. Now we have to calculate the probability that is covered by this set using the information given in the probability distribution. By step 2) of Example 1.1.9 we get that $x_0 = 0.5$ fulfills the inner function with a probability of 50%. As this is less then the required threshold of 90%, the choice $x_0 = 0.5$ is infeasible.

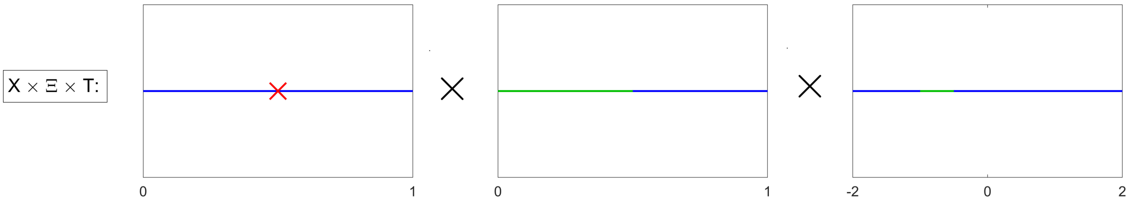


Figure 1.2: Fixed decision $x_0 \in X$ (red cross on the left), feasible realizations $\Omega(x_0) \subseteq \Xi$ (green line segment in the middle) and corresponding worst-case scenarios (green line segment on the right)

Last, but not least we enter step 3), where we use the gathered information to calculate a minimal $x^* \in [0, 1]$ satisfying $\varphi(x^*) \geq 0.9$. As the probability function can be simplified to $\varphi(x) = x$, we see that $x^* = 0.9$ is the optimal solution. This means that the corresponding set $\Omega(x^*)$ covers at least 90% of Ξ for all $t \in T$. This decision x^* with the corresponding uncertainty subset $\Omega(x^*) \times \bigcup_{\xi \in \Omega(x^*)} \{t^*(x^*, \xi)\} \subseteq \Xi \times T$ can be seen in Figure 1.3.

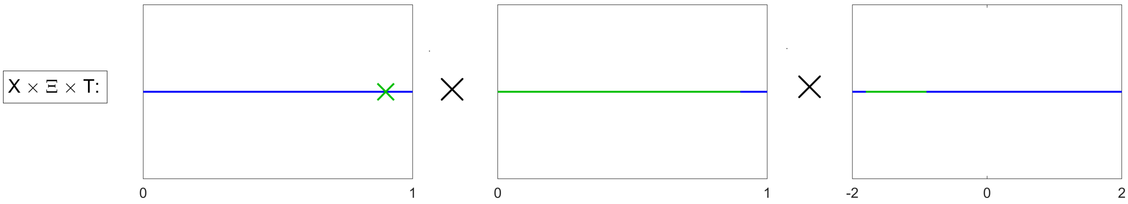


Figure 1.3: Optimal decision $x^* \in X$ (green cross on the left), set of feasible realizations $\Omega(x^*) \subseteq \Xi$ (green line segment in the middle) and corresponding worst-case scenarios (green line segment on the right)

So far, we presented the most important analytical results for standard probust optimization problems for this thesis. Additionally, we solved a probust optimization problem analytically and visualized its solution in the decision-realization-scenario space $X \times \Xi \times T$. Thereby, we are at the edge of current research. Before we start with numerical solving strategies for probust optimization problems, we introduce an extended problem definition that will allow us to model the problems in Part II more easily. To be able to formulate this extended problem definition, we need some more results from literature concerning decision-dependent uncertainty.

1.2 Decision-dependent uncertainty

The main goal of this section is to introduce decision-dependent uncertainties to generalize the definition of a probust optimization problem in the next section.
As decision-dependent uncertainty implies that the uncertainty set changes with the decision, we have to talk about point-to-set mappings before going into details with specialized concepts for semi-infinite and chance constrained optimization problems.
Unfortunately, the uncertainty concepts in these two optimization problem classes require slightly different continuity concepts. Therefore, we first introduce these concepts and afterwards discuss their relationship to each other.

Definition 1.2.1 (Correspondences)
Let X, Y be two sets, we call a mapping Γ from the set X to the power set of Y a correspondence. We will further denote correspondences as $\Gamma : X \rightrightarrows Y$. We call $dom(\Gamma) := \{x \in X \mid \Gamma(x) \neq \emptyset\}$ the domain of Γ and $gr(\Gamma) := \{(x, y) \in X \times Y \mid y \in \Gamma(x)\}$ the graph of Γ.
Γ is bounded if $\bigcup_{x \in X} \Gamma(x)$ is bounded.
We call Γ compact/convex-valued if for all $x \in X$ the set $\Gamma(x)$ is compact/convex.

Definition 1.2.2 (Upper/lower hemi-continuity of correspondences)
Given a correspondence $\Gamma : X \rightrightarrows Y$ between two topological spaces $(X, \mathcal{T}_X), (Y, \mathcal{T}_Y)$. We say that Γ is upper hemi-continuous (u.h.c.) at $\bar{x} \in X$, if the following holds

$$\forall V \in \mathcal{T}_Y \exists U \in \mathcal{U}(\bar{x}) : \Gamma(\bar{x}) \subseteq V \Rightarrow \Gamma(x) \subseteq V \ \forall x \in U$$

Γ *is lower hemi-continuous (l.h.c.) at* $\bar{x} \in X$, *if the following holds*

$$\forall V \in \mathcal{T}_Y \exists U \in \mathcal{U}(\bar{x}) : \Gamma(\bar{x}) \cap V \neq \emptyset \Rightarrow \Gamma(x) \cap V \neq \emptyset \ \forall x \in U$$

Γ *is continuous at* $\bar{x}$ *if it is lower and upper continuous at* $\bar{x} \in X$.
Γ *is (lower/upper hemi-)continuous if* Γ *is (lower/upper hemi-)continuous for all* $x \in X$.

Definition 1.2.3 (Inner/outer semi-continuity of correspondences)
Given a correspondence $\Gamma : X \rightrightarrows Y$ *between two topological spaces* $(X, \mathcal{T}_X), (Y, \mathcal{T}_Y)$. *We say that* Γ *is inner semi-continuous (i.s.c.) at* $\bar{x} \in X$, *if the following holds*

$$\liminf_{x \to \bar{x}} \Gamma(x) \supseteq \Gamma(\bar{x}).$$

Γ *is outer semi-continuous (o.s.c.) at* $\bar{x} \in X$, *if the following holds*

$$\limsup_{x \to \bar{x}} \Gamma(x) \subseteq \Gamma(\bar{x}).$$

More definitions referring to correspondences and structural properties of these can be found, e.g., in Berge's book [14], Chapter 16 of Aliprantis' book [6], Chapter 5 of Rockafeller's book [66] and a manuscript by Border [18].
These continuity concepts stem from different contexts, but are closely related as the following proposition shows:

Proposition 1.2.4 (U.h.c./l.h.c and o.s.c/i.s.c., Prop. 23 in [18])
Let $\Gamma : X \rightrightarrows Y$ *be a correspondence.*
Γ *is lower hemi-continuous at* $x \in X$ *if and only if it is inner semi-continuous at* x.
If Γ *has nonempty compact values, then* Γ *is upper hemi-continuous at* $x \in X$ *if and only if it is outer semi-continuous at* x.

Also connected to the introduced continuity concepts for correspondences is continuity w.r.t. the Hausdorff metric, when we interpret the mapping $f : X \to 2^Y$ as a correspondence with $\Gamma : X \rightrightarrows Y, x \mapsto f(x)$. We are interested in how continuity of Γ is related to continuity of f w.r.t. the topology on Y. Here the following theorem helps us:

Theorem 1.2.5 (U.h.c./l.h.c. and Hausdorff-continuity, Thm. 16.16 in [6])
Let $\Gamma : X \rightrightarrows Y$ *be a nonempty compact-valued correspondence from a topological space* $(X, \mathcal{T}_X)$ *into a metrizable space* (Y, d_Y), *and let* $\mathcal{K}(Y)$ *denote the space of nonempty compact subsets of* Y *endowed with its Hausdorff-metric topology. Then the function* $f : X \to \mathcal{K}(Y)$ *defined by* $f(x) = \Gamma(x)$ *is continuous if and only if the correspondence* Γ *is continuous.*

More details can be found, e.g., in Section 2.5 of the book by Göpfert et al. [35].
All together, the three different continuity concepts of upper/lower hemi-continuity, inner/outer semi-continuity and continuity w.r.t. the Hausdorff metric

$$d_H : \mathcal{K}(X) \times \mathcal{K}(X) \to \mathbb{R}, (C_1, C_2) \mapsto \inf\{\epsilon \geq 0 \mid C_1 \subseteq B_\epsilon(C_2) \wedge C_2 \subseteq B_\epsilon(C_1)\}$$

coincide if we consider nonempty, compact-valued correspondences.

Last, but not least, we are interested in the behavior of the function $f(x) = \max_{y\in\Gamma(x)} g(x, y)$ for continuous correspondences Γ. The most famous result in this context is Berge's maximum theorem from Berge's book [14], that we use here in three variants.

Lemma 1.2.6 (L.s.c. of supremum-function, Lem. 16.29 in [6])
Let $\Gamma : X \rightrightarrows Y$ be a lower hemi-continuous correspondence between topological spaces with nonempty values and let the function $g : gr(\Gamma) \to \mathbb{R}$ be lower semi-continuous. Define the extended real function $f : X \to \overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ by

$$f(x) = \sup_{y\in\Gamma(x)} g(x, y).$$

Then the function f is lower semi-continuous.

Lemma 1.2.7 (U.s.c. of supremum-function, Lem. 16.30 in [6])
Let $\Gamma : X \rightrightarrows Y$ be upper hemi-continuous correspondence between topological spaces with nonempty, compact values and let the function $g : gr(\Gamma) \to \mathbb{R}$ be upper semi-continuous. Define the extended real function $f : X \to \overline{\mathbb{R}}$ by

$$f(x) = \max_{y\in\Gamma(x)} g(x, y).$$

Then the function f is upper semi-continuous.

Theorem 1.2.8 (Berge's maximum theorem, Thm. 16.31 in [6]))
Let $\Gamma : X \rightrightarrows Y$ be a continuous correspondence between topological spaces with nonempty compact values and suppose $g : gr(\Gamma) \to \mathbb{R}$ is continuous. Define the "value function" $f : X \to \mathbb{R}$ by

$$f(x) = \max_{y\in\Gamma(x)} g(x, y)$$

and the correspondence $y^* : X \rightrightarrows Y$ of maximizers by

$$y^*(x) = \{y \in \Gamma(x) \mid g(x, y) = f(x)\}.$$

Then

1. *the value function f is continuous and*

2. *if either g has a continuous extension to all of $X \times Y$ or Y is Hausdorff, then the "argmax" correspondence y^*, is upper hemi-continuous with nonempty compact values.*

With this knowledge, we can start considering generalized semi-infinite optimization.

Generalized semi-infinite optimization

We now consider the most important results from generalized semi-infinite optimization which we will use later in this thesis. We interpret generalized semi-infinite optimization problems as the decision-dependent extension of robust optimization and semi-infinite optimization problems.
Generalized semi-infinite optimization problems have the form

$$\text{GSIP}: \min_{x\in X} f(x) \text{ s.t. } g(x,t) \leq 0 \ \forall t \in T(x) = \{t \in \mathbb{R}^q \mid u_j(x,t) \leq 0 \ \forall j \in J\},$$

where $X \subseteq \mathbb{R}^n$ is some set, J is a finite index set and $f : X \to \mathbb{R},\ g, u_j : X \times \mathbb{R}^q \to \mathbb{R}$ are functions for all $j \in J$ with $|T(x)| = \infty$ for all $x \in X$. If we compare this formulation with the semi-infinite problem introduced in Section 1.1, we see that the only difference is that the uncertainty set T now depends on the decision $x \in X$.
The existence of a solution of a GSIP is a by basic analysis and the Lemma 1.2.6:

Corollary 1.2.9 (Solution existence of GSIP)
Let X be a compact set, $f : X \to \mathbb{R}$ be lower semi-continuous, $T : X \rightrightarrows \mathbb{R}^q$ be a non-empty, lower hemi-continuous correspondence, $T_{\max} = \bigcup_{x\in X} T(x)$ and $g : X \times T_{\max} \to \mathbb{R}$ be a lower semi-continuous function w.r.t. $(x,t) \in X \times T_{\max}$. Then the induced GSIP has a well-defined solution $f^* \in \mathbb{R}$ and a well-defined minimizer $x^* \in X$, if the feasible set is not empty.

As we know that the correspondence $T : X \rightrightarrows \mathbb{R}^q$ in a GSIP is defined by inequality constraints $u_j : X \times \mathbb{R}^q \to \mathbb{R}, j \in J$, we have to ask which properties of $u_j, j \in J$ ensures the lower hemi-continuity of T.
Stein and Still give an answer to this in Lemma 2 in [77] as well as Chapter 3 of Stein's book [78] (see Lemma 3.2.2 and Proposition 3.2.27).

Lemma 1.2.10 (Sufficient condition for l.h.c, Lem. 2 in [77])
Given a GSIP with continuous functions $u_j : X \times Y \to \mathbb{R}$ for $j \in J$, the following holds for the correspondence $T : X \rightrightarrows Y, x \mapsto \{y \in Y \mid u_j(x,y) \leq 0 \ \forall j \in J\}$:

1. *If the functions $u(x,t) = Ax + Bt - b$ is (componentwise) affine linear then T is continuous w.r.t. $x \in X$.*

2. *Let $U \subseteq X$ be open. Let for any $x \in U, j \in J$ the function $u_j(x,\cdot)$ be convex in t and let for any $x \in U$ the Slater condition hold: There exists $\bar{t} \in \mathbb{R}^m$ such that $u_j(x,\bar{t}) < 0$ for all $j \in J$. Then T is continuous w.r.t. $x \in U$.*

3. *Let $U \subseteq X$ be open. Let for any $x \in U$ the condition (MFCQ) be fulfilled at all $t \in T(x)$. Then T is continuous w.r.t. $x \in U$.*

The Mangasarian-Fromovitz constraint qualification (MFCQ) is fulfilled at a fixed point $(x,t) \in X \times \mathbb{R}^q$ if

$$\exists d \in \mathbb{R}^m : \langle \nabla_t u_j(x,t), d\rangle < 0 \ \forall j \in J_0(x) = \{j \in J \mid u_j(x,t) = 0\}.$$

To be able to ensure upper hemi-continuity of a correspondence described by inequality constraints, we use the following theorem by Hogan. Please note that we are allowed to exchange the original "closedness of Γ" by "outer semi-continuity of Γ" by means of Lemma 5.7*a* of Rockafellar's book [66].

Theorem 1.2.11 (Sufficient condition for o.s.c., Thm. 10 in [46])
Consider two sets topological space X, Y and functions $u_j : X \times Y \to \mathbb{R}, j \in J$, where J is a finite index set, that are lower semi-continuous on $X \times Y$. Define the correspondence $\Gamma : X \rightrightarrows Y$ as

$$\Gamma(x) := \{y \in Y \mid u_j(x,y) \leq 0 \ \forall j \in J\}.$$

Then Γ is outer semi-continuous and closed-valued.

Proposition 1.2.4 implies now that lower semi-continuity of u_j for all $j \in J$ together with the boundedness of $\bigcup_{x \in X} \Gamma(x)$ guarantee that Γ is upper hemi-continuous w.r.t. $x \in X$. We can also talk about an unique solution of a GSIP using:

Lemma 1.2.12 (Convex feasible set of GSIP, Lem. 2 in [82])
Suppose that the function $g : X \times Y \to \mathbb{R}$ is convex in $(x,y) \in X \times Y$ and assume that the following set-valued inclusion holds for the correspondence $\Gamma : X \rightrightarrows Y$:
For any $x_1, x_2 \in X$ and $\lambda \in (0,1)$ it holds

$$\Gamma(\lambda x_1 + (1-\lambda)x_2) \subseteq \lambda\Gamma(x_1) + (1-\lambda)\Gamma(x_2).$$

Then, the feasible set of GSIP is convex.

Please note that the original proof still works if we substitute the convexity of $g(\cdot, t)$ w.r.t. $x \in X$ by quasi-convexity.

Next to this analytical properties of GSIPs, we are also interested in solving these problems numerically. In this context the lower-level problem similar to (1.1) is defined by fixing some decision $\overline{x} \in \mathbb{R}^n$:

$$Q(\overline{x}) \ : \ \max_{t \in T(\overline{x})} g(\overline{x}, t) \tag{1.4}$$

Numerical solution approaches for GSIPs can be categorized the same way as solution approaches for SIPs in the last section. We recommend the survey by Vazquez et al. [84] and refer to the monograph [78] for more details.

As discretization methods suffer from the changing set $T(x)$ for different $x \in X$, one could think that they get less attention. Nevertheless, in publications by Schwientek et al. (see [71, 72]) a transformation based discretization method is introduced which deals with this problem.

The transformation based discretization method uses a transformation $\mathcal{T} : X \times Z \to \mathbb{R}^q$ that satisfies $\mathcal{T}(x, Z) = T(x)$ for all fixed $x \in X$.

Denoting $\tilde{g} : X \times Z \to \mathbb{R}, (x,z) \mapsto \tilde{g}(x,z) := g(x, \mathcal{T}(x,z))$, we can write it down as:

Algorithm 2 Transformation based discretization method from Schwientek [72]

```
Inputs:
    Generalized semi-infinite problem instance GSIP, starting decision
    $x_0 \in X$, reference set $Z$, starting discretization $Z_0 \subseteq Z$, transformation
    $\mathcal{T} : X \times Z \to \mathbb{R}^q$
Initialize:
    $k := 0$
do
    $x_{k+1} \leftarrow \arg\min_{x \in X} f(x)$ s.t. $\tilde{g}(x, z) \leq 0 \ \forall z \in Z_k$
    $t_k^* \leftarrow \arg\max_{t \in T(x_{k+1})} g(x_{k+1}, t)$
    $z_k^* \leftarrow \mathcal{T}(x_{k+1}, z_k^*) = t_k^*$
    $Z_{k+1} \leftarrow Z_k \cup \{z_k^*\}$
    $k \leftarrow k + 1$
while $g(x_k, t_k^*) > 0$
Results:
    Sequence $(x_k, Z_k)_{k \in \mathbb{N}}$
```

This algorithm guarantees that all accumulation points of $(x_k)_{k \in \mathbb{N}}$ are minimizers w.r.t. the original GSIP if $f, g, \mathcal{T}$ are continuous and X, Z, T are compact(-valued) (see Theorem 7.3.2 in [72]). Those assumptions are mild considering Corollary 1.2.9.

Decision-dependent chance constrained optimization

Next, we deal with decision-dependent chance constrained optimization problems. The difference to chance constrained optimization problems is that the probability distribution now depends on the decision.
Such problems can be noted as

$$\text{DDCC} : \min_{x \in X} f(x) \text{ s.t. } \mathbb{P}(x)(g(x, \xi) \leq 0) \geq p,$$

where $X \subseteq \mathbb{R}^n$ is some set, $(\Xi, \mathcal{A}, \mathbb{P}(x))$ is a measurable space for all $x \in X$, $p \in [0, 1]$ is a probability threshold, $f : X \to \mathbb{R}$ is a continuous function, g is a Caratheodory-function and $\mathbb{P} : X \to \mathcal{P}(\Xi, \Omega)$ is a function that maps into the set of probability measures on $(\Xi, \mathcal{A})$. If we compare this formulation with the chance constrained optimization problem introduced in Section 1.1, we see that the only difference is that the distribution function $\mathbb{P}$ depends now on the decision $x \in X$.
The existence of a solution of a DDCC is not known in general to the best of our knowledge. Therefore, we introduce some definitions to be able to create a context where DDCC has a well-defined solution to close that gap. As a first step we have to define continuity w.r.t. a probability distribution. To define a quite general concept of continuity, we are inspired by a paper by Gibbs et al. [33]. There it was shown, that the discrepancy-metric can be bounded from above by a lot of other probability measures that are used more frequently, i.e., the total variation distance or the Kolmogorov metric.

Definition 1.2.13 ((Semi-)continuity w.r.t. discrepancy metric d_D)
Let (X, d_X) be a metric space, $(\Xi, \mathcal{A})$ be a measurable space and $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ be a function that maps into the set of probability measures on $(\Xi, \mathcal{A})$.
For $\mathcal{S} \subseteq \mathcal{A}$ we define the $\mathcal{S}$-discrepancy metric $d_{D(\mathcal{S})}$ as

$$d_{D(\mathcal{S})} : \mathcal{P}(\Xi, \mathcal{A}) \times \mathcal{P}(\Xi, \mathcal{A}) \to [0, 1], (\mathbb{P}_1, \mathbb{P}_2) \mapsto \sup_{A \in \mathcal{S}} |\mathbb{P}_1(A) - \mathbb{P}_2(A)|.$$

We call $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ upper semi-continuous at $\overline{x} \in X$ w.r.t. the $\mathcal{S}$-discrepancy metric if for all $\epsilon > 0$ there exists a $\delta > 0$ such that

$$\sup_{A \in \mathcal{S}} \mathbb{P}(\bar{x})(A) - \mathbb{P}(x)(A) < \epsilon \ \forall x \in B_\delta(\bar{x}).$$

We call $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ lower semi-continuous at $\overline{x} \in X$ w.r.t. the $\mathcal{S}$-discrepancy metric if for all $\epsilon > 0$ there exists a $\delta > 0$ such that

$$\sup_{A \in \mathcal{S}} \mathbb{P}(x)(A) - \mathbb{P}(\bar{x})(A) < \epsilon \ \forall x \in B_\delta(\bar{x}).$$

We call $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ continuous at $\overline{x} \in X$ w.r.t. the $\mathcal{S}$-discrepancy metric if it is upper and lower semi-continuous w.r.t. the $\mathcal{S}$-discrepancy metric.
We call $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ (upper/lower semi-)continuous w.r.t. the $\mathcal{S}$-discrepancy metric if it is (upper/lower semi-)continuous for all $x \in X$.
If we choose $\mathcal{S} = \mathcal{A}$, we do not use the prefix $\mathcal{S}$ for the continuity descriptions, but write continuous with respect to the discrepancy-metric d_D.

It is noted by Henrion et al. in [41] that the $\mathcal{S}$-discrepancy metric $d_{D(\mathcal{S})}$ is a semi-metric on $(\mathcal{P}(\Xi, \mathcal{A})$ for any $\mathcal{S} \subseteq \mathcal{A}$. There is also noted that it is a metric, if $\mathcal{S}$ contains a generator of $\mathcal{A}$. For example, if we are interested in the Borel-σ-algebra of $\Xi = \mathbb{R}^m$, we could choose the set of all closed sets or all hyper-boxes $(-\infty, \xi], \xi \in \Xi$ as a generator. Since we are just interested in theoretical results, we choose $\mathcal{S} = \mathcal{A}$ in the remainder of this chapter.
Some examples of probability distribution functions that are continuous w.r.t. the discrepancy metric d_D are given next:

Example 1.2.14 (d_D-continuous mappings)
(i) We consider the probability measures on $(\mathbb{R}, \mathcal{B})$ that are induced by the uniform distributed random variable $Z \sim \mathcal{U}([a(x), b(x)])$, where $x \in X \subseteq \mathbb{R}^n, n \in \mathbb{N}$. X is supposed to be a compact set, $a, b : X \to \mathbb{R}$ are continuous w.r.t. x and $m := \inf_{x \in X} b(x) - a(x) > 0$ is fulfilled. Then the function $\mathbb{P} : X \to \mathcal{P}(\mathbb{R}, \mathcal{B})$ is continuous w.r.t. d_D.
(ii) We consider the probability measures on $(\mathbb{R}, \mathcal{B})$ that are induced by the Gaussian distributed random variable $Z \sim \mathcal{N}(\mu(x), \sigma(x))$, where $x \in X \subseteq \mathbb{R}^n, n \in \mathbb{N}$. We assume that the functions $\mu : X \to \mathbb{R}, \sigma : X \to \mathbb{R}_{\geq 0}$ are continuous w.r.t. x and fulfill $m := \inf_{x \in X} \sigma(x) > 0$. Then the function $\mathbb{P} : X \to \mathcal{P}(\mathbb{R}, \mathcal{B})$ is continuous w.r.t. d_D.

Another property of this metric is given by the following theorem which connects it to weak convergence of probability measures:

Theorem 1.2.15 (Weak convergence and d_D-continuity, Thm. 2 in [16])
Let $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, Ξ be locally connected, $\mathcal{S} \subseteq \mathcal{A}$ and $(\mathbb{P}_n)_{n\in\mathbb{N}} \subseteq \mathcal{P}(\Xi, \mathcal{A})$ be a weakly convergent sequence towards $\mathbb{P}$. Then

$$\lim_{n\to\infty} d_{D(\mathcal{S})}(\mathbb{P}_n, \mathbb{P}) = 0 \Leftrightarrow \lim_{\epsilon\to 0} \sup_{A\in\mathcal{S}} \mathbb{P}(B_\epsilon(\partial A)) = 0.$$

This means that weak convergence of probability measures can be upgraded to convergence w.r.t. the discrepancy-metric if all boundaries are null sets w.r.t. the weak limit probability measure.

As we understand the concept of continuity w.r.t. the discrepancy-metric better now, we are interested which context is sufficient to guarantee that DDCCs have well-defined solutions.

The survey about decision-dependent stochastic optimization by Hellemo et al. [40] does not comment on such statements. Decision-dependent probabilities with continuous distributions are just barely mentioned. Consequently, we close that gap inspired by the following theorem of Beer and Villar:

Theorem 1.2.16 (Upper semi-continuity of measures, Thm. 3.2 in [10])
Let μ be a measure on the Borel-σ-algebra $\mathcal{B}$ of a metric space (X, d_X).

1. *If μ is locally finite, then μ is upper semi-continuous w.r.t. the Hausdorff-metric at each compact set $K \in \mathcal{K}(X)$.*
2. *If $A \subseteq X$ is closed and $\mu(A^C) < \infty$, then μ is upper semi-continuous w.r.t. the Hausdorff-metric at A.*

Together with Theorem 1.2.5 we conclude that every nonempty, compact-valued, continuous correspondence Γ implies the upper semi-continuity of $\mathbb{P}(\Gamma(\cdot)) : X \to [0, 1]$.

We reduce the assumptions to guarantee the upper semi-continuity of $\mathbb{P} \circ \Gamma$ by the next lemma.

Lemma 1.2.17 (Upper semi-continuity of $\mathbb{P} \circ \Gamma$)
Let (X, d_X) be a metric set, $(\Xi, \mathcal{B})$ be a Borel-measurable space, $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{B})$ be a mapping into the set of probability measures on $(\Xi, \mathcal{B})$ and $\Gamma : X \rightrightarrows \Xi$ be a correspondence satisfying $\Gamma(x) \in \mathcal{B}$ for all $x \in X$.
(i) If $\mathbb{P}$ is upper semi-continuous w.r.t. the discrepancy metric d_D and $\Gamma : X \rightrightarrows \Xi$ is outer semi-continuous, then the function $f : X \to [0, 1], x \mapsto \mathbb{P}(x)(\Gamma(x))$ is upper semi-continuous.
(ii) If $\mathbb{P}$ is lower semi-continuous w.r.t. the discrepancy metric d_D and $\Gamma : X \rightrightarrows \Xi$ is inner semi-continuous, then the function $f : X \to [0, 1], x \mapsto \mathbb{P}(x)(\Gamma(x))$ is lower semi-continuous.

Proof. We show both claims directly using an ϵ-δ-criterion:
As we want to show that f is upper semi-continuous, we have to show for any $x \in X$ and any $\epsilon > 0$ there exists a $\delta > 0$ such that for all $y \in B_\delta(x)$ it holds $f(y) - f(x) < \epsilon$.

As Γ is outer semi-continuous, we know that $\limsup_{y\to x}\Gamma(y) \subseteq \Gamma(x)$ and by Fatou's lemma $\limsup_{y\to x}\mathbb{P}(z)(\Gamma(y)) \leq \mathbb{P}(z)(\Gamma(x))$ for all $z \in X$. Consequently, the function

$$f_z : X \to [0,1], x \mapsto \mathbb{P}(z)(\Gamma(x))$$

is upper semi-continuous at x for any $z \in X$. This means that especially for the choice $z = x$ we can guarantee that there exists a $\delta_1 > 0$ such that $\mathbb{P}(x)(\Gamma(y)) - \mathbb{P}(x)(\Gamma(x)) < \frac{\epsilon}{2}$ for all $y \in B_{\delta_1}(x)$.
Additionally, with the upper semi-continuity of $\mathbb{P}$ w.r.t. d_D, we know that there exists an $\delta_2 > 0$ such that for all $y \in B_{\delta_2}(x)$ we can guarantee $\sup_{A\in\mathcal{B}}\mathbb{P}(x)(A) - \mathbb{P}(y)(A) < \frac{\epsilon}{2}$. All together, we can estimate for $y \in B_\delta(x)$, where $\delta := \min\{\delta_1, \delta_2\}$:

$$\begin{aligned} f(y) - f(x) &= \mathbb{P}(y)(\Gamma(y)) - \mathbb{P}(x)(\Gamma(x)) \\ &= \mathbb{P}(y)(\Gamma(y)) - \mathbb{P}(x)(\Gamma(y)) + \mathbb{P}(x)(\Gamma(y)) - \mathbb{P}(x)(\Gamma(x)) \\ &\leq \sup_{A\in\mathcal{B}}\{\mathbb{P}(y)(A) - \mathbb{P}(x)(A)\} + \mathbb{P}(x)(\Gamma(y)) - \mathbb{P}(x)(\Gamma(x)) \\ &< \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \end{aligned}$$

As $x \in X$ was chosen arbitrarily, the first part of the claim holds.
The second part of the claim can be shown the same way using the lower semi-continuity of $\mathbb{P}$, Fatou's lemma which implies $\mathbb{P}(z)(\liminf_{y\to x}\Gamma(y)) \leq \liminf_{y\to x}\mathbb{P}(z)(\Gamma(y))$ for each $z \in Z$ and the inner semi-continuity of Γ. □

Using this lemma in the context of decision-dependent chance constrained optimization problems we can state a sufficient condition for a well-defined solution. To formulate this result we need to guarantee that the feasible set of a DDCC is not empty. Therefore, we calculate the following probability threshold

$$p_{\max} = \max_{x\in X, p\in[0,1]} p \text{ s.t. } \mathbb{P}(x)(g(x,\xi) \leq 0) \geq p \tag{1.5}$$

and state:

Corollary 1.2.18 (Solution existence of DDCC)
Let (X, d_X) be a metric space, $(\Xi, \mathcal{A})$ be a measurable space, $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ be an upper semi-continuous function w.r.t. the discrepancy metric d_D and $g : X \times \Xi \to \mathbb{R}$ be a lower semi-continuous function satisfying $\Gamma(x) := \{\xi \in \Xi \mid g(x,\xi) \leq 0\} \in \mathcal{A}$ for all $x \in X$. Then the function $\varphi : X \to [0,1], x \mapsto \mathbb{P}(x)(\Gamma(x))$ is upper semi-continuous.
If additionally the set X is compact, $f : X \to \mathbb{R}$ is lower semi-continuous and $p \leq p_{\max}$, then the decision-dependent optimization problem has a well-defined minimizer $x^* \in X$.

Proof. We show this claim directly.
By Theorem 1.2.11 we know that Γ is an outer semi-continuous correspondence w.r.t. $x \in X$. By Lemma 1.2.17 we know that the function $\varphi : X \to [0,1], x \mapsto \mathbb{P}(x)(\Gamma(x))$ is upper semi-continuous w.r.t. $x \in X$. Consequently, the feasible set of the DDCC is closed and compact as X is compact. This implies that the optimization problem defined to calculate $p_{\max}$ is well-defined as well. Because $p \leq p_{\max}$ we know that the feasible set is not empty and due to the lower semi-continuity of f we know that there exists a minimizer $x^* \in X$. □

After focusing on the existence of a solution, we can study conditions to guarantee an unique solution of a DDCC. To ensure convexity of the feasible set of a DDCC, we use an extended version of the concept that is introduced for joint chance constrained optimization problems in Definition 1.1.4:

Definition 1.2.19 (α-concave probability distribution functions)
Let $X \subseteq \mathbb{R}^n$ be any convex set and $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ be a mapping from X into the set of probability measures $\mathcal{P}(\Xi, \mathcal{A})$ on the measurable space $(\Xi, \mathcal{A})$. We call $\mathbb{P}$ an α-concave probability distribution function, if for all $x_1, x_2 \in X, A_1, A_2 \in \mathcal{A}, \lambda \in [0, 1]$ we can guarantee:

$$\mathbb{P}(x(\lambda))(A(\lambda)) \geq m_\alpha(\mathbb{P}(x_1)(A_1), \mathbb{P}(x_2)(A_2), \lambda),$$

where $x(\lambda) = \lambda x_1 + (1-\lambda)x_2$, $A(\lambda) = \lambda A_1 + (1-\lambda)A_2$ and $m_\alpha : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times [0, 1] \to \mathbb{R}_{\geq 0}$ defined as in Definition 1.1.4 for fixed $\alpha \in \mathbb{R}$.

If $\mathbb{P}$ is independent w.r.t. $x \in X$, then sufficient conditions for φ to be an α-concave function are stated in Theorem 4.39 from Shapiro's book [75].
With this concavity concept in mind, we state the following theorem which can be used to ensure the convexity of the feasible set of a DDCC.

Theorem 1.2.20 (Concavity of φ)
Let (X, d_X) be a metric space, X and $\Xi \subseteq \mathbb{R}^m$ be convex sets, $(\Xi, \mathcal{A})$ be a measurable space, $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ be an α-concave probability distribution function and $\Gamma : X \rightrightarrows \Xi$ be a correspondence that satisfies $\Gamma(x(\lambda)) \supseteq \lambda\Gamma(x_1) + (1 - \lambda)\Gamma(x_2)$ for all $x_1, x_2 \in X, \lambda \in [0, 1], x(\lambda) = \lambda x_1 + (1 - \lambda)x_2$ and $\Gamma(x) \in \mathcal{A}$ for all $x \in X$. Then the function $\varphi : X \to [0, 1], x \mapsto \mathbb{P}(x)(\Gamma(x))$ is α-concave.

Proof. We show this claim directly by using the definition of an α-concave probability distribution function.
Let us fix arbitrary $x_1, x_2 \in X, \lambda \in [0, 1]$ and define $x(\lambda) := \lambda x_1 + (1 - \lambda)x_2$.
By assumption Γ satisfies $\Gamma(x(\lambda)) \supseteq \lambda\Gamma(x_1) + (1 - \lambda)\Gamma(x_2)$ and consequently we can estimate using the α-concavity of $\mathbb{P}$:

$$\begin{aligned}
\varphi(x(\lambda)) &= \mathbb{P}(x(\lambda))(\Gamma(x(\lambda))) \\
&\geq \mathbb{P}(x(\lambda))(\lambda\Gamma(x_1) + (1 - \lambda)\Gamma(x_2)) \\
&\geq m_\alpha(\mathbb{P}(x_1)(\Gamma(x_1)), \mathbb{P}(x_2)(\Gamma(x_2)), \lambda) \\
&= m_\alpha(\varphi(x_1), \varphi(x_2), \lambda)
\end{aligned}$$

Consequently, because $x_1, x_2 \in X, \lambda \in [0, 1]$ were chosen arbitrarily, φ is α-concave and the claim holds. □

The next corollary states a sufficient condition for when a correspondence satisfies the convexity assumption of the last theorem:

Corollary 1.2.21 (Sufficient condition for convex Γ)
Let $X \subseteq \mathbb{R}^n, Y \subseteq \mathbb{R}^m$ be convex sets and $\Gamma : X \rightrightarrows Y$ be a correspondence defined by $\Gamma(x) = \{y \in Y \mid g(x,y) \leq 0\}$ with quasi-convex function $g : X \times Y \to \mathbb{R}$ w.r.t. $(x,y) \in X \times Y$. Then Γ satisfies for any $x_1, x_2 \in X, \lambda \in [0,1]$ and $x(\lambda) := \lambda x_1 + (1-\lambda)x_2$ the relation

$$\Gamma(x(\lambda)) \supseteq \lambda\Gamma(x_1) + (1-\lambda)\Gamma(x_2).$$

Proof. We show this claim directly by using the quasi-convexity of g:
Let us fix arbitrary $x_1, x_2 \in X, \lambda \in [0,1]$ to define $x(\lambda) = \lambda x_1 + (1-\lambda)x_2$ and take arbitrary $y_1 \in \Gamma(x_1), y_2 \in \Gamma(x_2)$. We show that $y(\lambda) = \lambda y_1 + (1-\lambda)y_2 \in \Gamma(x(\lambda))$.
By the quasi-convexity of g we can estimate

$$g(x(\lambda), y(\lambda)) \leq \max\{g(x_1,y_1), g(x_2,y_2)\} \leq 0.$$

As $y_1 \in \Gamma(x_1), y_2 \in \Gamma(x_2)$ were chosen arbitrarily, we know that the set inclusion $\Gamma(x(\lambda)) \supseteq \lambda\Gamma(x_1) + (1-\lambda)\Gamma(x_2)$ holds. Since $x_1, x_2 \in X$ and $\lambda \in [0,1]$ were chosen arbitrarily, the statement is proven. □

With these concepts, we can now extend the definition of (standard) probust optimization problems and consider well-defined solutions in this framework.

1.3 Generalized probust optimization problems

In the last sections we introduced semi-infinite optimization and joint chance constrained optimization problems as well as decision-dependent uncertainty concepts. Before extending the definition of (standard) probust optimization problems from Section 1.1, we introduce so called *probust terms.*

While we can consider semi-infinite optimization problems as probust optimization problems with fixed realization $\xi \in \Xi$ and chance constrained optimization problems as probust optimization problems with fixed scenario $t \in T$, these probust terms correspond to standard probust optimization problems with fixed decision $x \in X$. Analyzing these terms here and in Chapter 2 provides new insights which are crucial to handle probust optimization problems.

In this section, we define probust terms and discuss continuity and convexity properties of parametrized probust terms that we interpret as probust functions.

With this knowledge, we can focus on solution existence and uniqueness of generalized probust optimization problems.

We close this chapter by proving that generalized probust optimization problems can be reduced to standard probust optimization problems, if appropriate transformations of the corresponding uncertainty sets are given. We comment on how the solution of the original generalized probust optimization problem transforms in this context.

(Generalized) Probust terms

As explained, probust terms are the missing piece between semi-infinite optimization problems, joint chance constrained optimization problems and (standard) probust optimization problems. We now introduce a standard and a general form of these terms and show under which assumptions they are well-defined.

Definition 1.3.1 (Probust term)
Let $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $T : \Xi \rightrightarrows \mathbb{R}^q$ be a correspondence, $T_{\max} := \bigcup_{\xi \in \Xi} T(\xi)$ and $g : \Xi \times T_{\max} \to \mathbb{R}$ be a function that satisfies that the set of feasible realizations $\Omega := \{\xi \in \Xi \mid g(\xi, t) \leq 0 \ \forall t \in T(\xi)\}$ is measurable w.r.t. $\mathcal{A}$. Then we call

$$\varphi := \mathbb{P}(g(\xi, t) \leq 0 \ \forall t \in T(\xi)) \in [0, 1]$$

general probust term.
We call φ standard probust term, if T is constant.

Because later we are interested in sets of feasible realizations which are defined by finite many scenarios instead of the whole scenario set T, we make the following notation for convenience:

Notation
Given a probust term φ, a scenario $t \in T_{\max}$ and a subset $S \subseteq T_{\max}$, we define

$$\begin{aligned} \Omega(\emptyset) &:= \Xi, \\ \Omega(t) &:= \{\xi \in \Xi \mid g(\xi, t) \leq 0\}, \\ \Omega(S) &:= \bigcap_{t \in S} \Omega(t). \end{aligned}$$

Additionally to this last notation, it will come in handy to be able to represent probust terms $\varphi \in [0, 1]$ in several ways. This allows us to use the representation that fits the current analysis or application best. We can rewrite the value of a probust term as

$$\varphi = \mathbb{P}(g(\xi, t) \leq 0 \ \forall t \in T(\xi)) = \mathbb{P}\left(\sup_{t \in T(\xi)} g(\xi, t) \leq 0 \right) = \mathbb{P}(\Omega).$$

If we consider standard probust terms, we can also write

$$\varphi = \mathbb{P}(\Omega(T_{\max})) = \mathbb{P}(\Omega(T)).$$

Be aware that $\Omega(T_{\max}) \neq \{\xi \in \Xi \mid \sup_{t \in T(\xi)} g(\xi, t)\}$ in general.
Moreover, the assumption $\Omega \in \mathcal{A}$ in Definition 1.3.1 is not always fulfilled and not easy to check directly. Therefore, we are interested in a way to guarantee the measurability of Ω by considering $\Omega(t), t \in T_{\max}$.

A sufficient condition that guarantees well-defined probust terms can be found in the next proposition. Because its assumptions hold for the application part of this thesis, it is quite useful for us.

Proposition 1.3.2 (Sufficient condition for measurable Ω)
Let $(\mathbb{R}^m, \mathcal{B}, \mathbb{P})$ be a probability space with Borel-σ-algebra, let $T : \Xi \rightrightarrows \mathbb{R}^q$ be a lower hemi-continuous correspondence and let $g : \Xi \times T_{\max} \to \mathbb{R}$ be a lower semi-continuous function w.r.t. $(\xi, t) \in \Xi \times T_{\max}$, then the probust term $\mathbb{P}(g(\xi, t) \leq 0 \ \forall t \in T(\xi))$ is well-defined.

Proof. We show this claim directly by Lemma 1.2.6.
Because the assumptions of Lemma 1.2.6 are fulfilled, we know that

$$g(\cdot, T) : \Xi \to \mathbb{R}, \xi \mapsto \sup_{t \in T(\xi)} g(\xi, t)$$

is lower semi-continuous w.r.t. $\xi \in \Xi$. Consequently, the set Ω is closed as the pre-image of $(-\infty, 0]$ under a lower semi-continuous function. Therefore, it is Borel-measurable and the claim holds. □

If we consider standard probust terms with a compact set of scenarios T, we know that the correspondence is constant, consequently lower hemi-continuous and thereby we just have to ensure the lower semi-continuity of g on $\Xi \times T$ to use the last proposition.

(Generalized) Probust functions

In the next step, we focus on parametrized probust terms which we interpret as probust functions. We introduce a standard and a general form of these functions and show which conditions are sufficient for (upper semi-)continuity of them. After that, we reduce the general form to the standard form by introducing appropriate transformations. We also give sufficient conditions for concavity of these functions.

Definition 1.3.3 (Probust function)
Let X be a set, $(\Xi, \mathcal{A})$ a measurable space, $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ a decision dependent probability distribution function, $T : X \times \Xi \rightrightarrows \mathbb{R}^q$ a correspondence, $T_{\max} := \bigcup_{x \in X, \xi \in \Xi} T(x, \xi)$ and $g : X \times \Xi \times T_{\max} \to \mathbb{R}$ an inner function such that the set of feasible realization satisfies $\Omega(x) := \{\xi \in \Xi \mid g(x, \xi, t) \leq 0 \ \forall t \in T(x, \xi)\} \in \mathcal{A}$ for all $x \in X$. Then we call

$$\varphi : X \to [0, 1], x \mapsto \varphi(x) := \mathbb{P}\,(x)\,(g(x, \xi, t) \leq 0 \ \forall t \in T(x, \xi))$$

general probust function.
If $\mathbb{P}$ and T are constant, we call φ standard probust function.

Since we aim to solve probust optimization problems, we are interested under which assumptions a solution exists and when this solution is unique. These properties of a solution are closely connected to the continuity and the concavity of the probust function that is used to describe the probust optimization problem. We start with sufficient conditions to guarantee their upper semi-continuity.

Proposition 1.3.4 (Upper semi-continuity of φ)
Let (X, d_X) be a metric space, $(\Xi, \mathcal{A})$ be a measurable space and $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ be a probability distribution function that is upper semi-continuous w.r.t. the discrepancy-metric d_D. Furthermore, let $T : X \times \Xi \rightrightarrows \mathbb{R}^q$ be a compact-valued, lower hemi-continuous correspondence and $g : X \times \Xi \times T_{\max} \to \mathbb{R}$ be a lower semi-continuous inner function w.r.t. $(x, \xi, t) \in X \times \Xi \times T_{\max}$ that satisfies $\Omega(x) \in \mathcal{A}$ for all $x \in X$, then the probust function $\varphi : X \to [0,1]$ is upper semi-continuous.

Proof. We show the using Lemma 1.2.6, Theorem 1.2.11 and Lemma 1.2.17:
We can use Lemma 1.2.6 since g is lower semi-continuous and T is lower hemi-continuous, which induces that

$$g(x, \xi, T) = \sup_{t \in T(x,\xi)} g(x, \xi, t)$$

is lower semi-continuous w.r.t. $(x, \xi) \in X \times \Xi$.
Because of Theorem 1.2.11 and Lemma 1.2.17 we know that the correspondence $\Omega(x) := \{\xi \in \Xi \mid g(x, \xi, T) \leq 0\}$ is upper hemi-continuous and therefore

$$\varphi : X \to [0,1], x \mapsto \mathbb{P}(x) \left(\sup_{t \in T(x,\xi)} g(x, \xi, t) \leq 0 \right)$$

is an upper semi-continuous function w.r.t. $x \in X$. □

To be able to ensure lower semi-continuity of φ, we have to strengthen our assumptions:

Proposition 1.3.5 (Continuity of φ)
Let (X, d_X) be a metric space, $(\Xi, \mathcal{A})$ be a measurable space and $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ be a probability distribution function that is continuous w.r.t. d_D. Furthermore, let the correspondence $T : X \times \Xi \rightrightarrows \mathbb{R}^q$ be a continuous and compact-valued and the inner function $g : X \times \Xi \times T_{\max} \to \mathbb{R}$ be continuous w.r.t. $(x, \xi, t) \in X \times \Xi \times T_{\max}$ satisfying $\Omega(x) \in \mathcal{A}$ and $\mathbb{P}(x)(\max_{t \in T(y,\xi)} g(y, \xi, t) = 0) = 0$ for all $x, y \in X$, then $\varphi : X \to [0,1]$ is a continuous function.

Proof. We show the claim by using basic analysis, Berge's maximum theorem, as well as Proposition 1.1.6, Proposition 1.1.7 and an ϵ-δ-approach:
Using Berge's maximum theorem 1.2.8, we know that

$$g(x, \xi, T) := \max_{t \in T(x,\xi)} g(x, \xi, t)$$

is well-defined and continuous w.r.t. $(x, \xi) \in X \times \Xi$.
Next we estimate for $x, y \in X$:

$$\begin{aligned}
|\varphi(x) - \varphi(y)| &= |\mathbb{P}(x)(g(x, \xi, T) \leq 0) - \mathbb{P}(y)(g(y, \xi, T) \leq 0)| \\
&\leq |\mathbb{P}(x)(g(x, \xi, T) \leq 0) - \mathbb{P}(x)(g(y, \xi, T) \leq 0)| \\
&\quad + |\mathbb{P}(x)(g(y, \xi, T) \leq 0) - \mathbb{P}(y)(g(y, \xi, T) \leq 0)| \\
&\leq |\mathbb{P}(x)(g(x, \xi, T) \leq 0) - \mathbb{P}(x)(g(y, \xi, T) \leq 0)| \\
&\quad + \sup_{A \in \mathcal{A}} |\mathbb{P}(x)(A) - \mathbb{P}(y)(A)|
\end{aligned}$$

Let us fix an arbitrary $\epsilon > 0$ under consideration of Proposition 1.1.6 and Proposition 1.1.7 with (fixed) distribution $\mathbb{P}(x)$. Then, we can estimate the first term in the last inequality. Additionally, we can estimating the second term by the continuity of $\mathbb{P}$ w.r.t. d_D to find $\delta_1, \delta_2 > 0$ such that

$$|\mathbb{P}(x)(g(x,\xi,T) \leq 0) - \mathbb{P}(x)(g(y,\xi,T) \leq 0)| < \frac{\epsilon}{2} \; \forall y \in B_{\delta_1}(x) \text{ and}$$
$$\sup_{A \in \mathcal{A}} |\mathbb{P}(x)(A) - \mathbb{P}(y)(A)| < \frac{\epsilon}{2} \; \forall y \in B_{\delta_2}(x).$$

Consequently $\delta := \min\{\delta_1, \delta_2\}$ and $y \in B_\delta(x)$ implies

$$|\varphi(x) - \varphi(y)| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon.$$

As $x \in X$ and $\epsilon > 0$ were chosen arbitrarily, the claim holds. □

Please note that in the standard setting $\mathbb{P}$ and T are constant. Consequently, we just have to guarantee that the inner function g is lower semi-continuous to conclude the upper semi-continuity of the probust function φ. Furthermore, we need a continuous function g and the condition $\mathbb{P}(\max_{t \in T} g(x,\xi,t) = 0) = 0$ for all $x \in X$ to ensure continuity of φ. These are exactly the results from Farshbaf-Shaker et al. [30].

Next to the continuity of a probust function we are interested in its concavity. The next theorem helps us:

Theorem 1.3.6 (Concavity of φ)
Let $X \subseteq \mathbb{R}^n, \Xi \subseteq \mathbb{R}^m$ be convex sets, $(\Xi, \mathcal{A})$ a measurable space, $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ an α-concave probability distribution, $g : X \times \Xi \times T_{\max} \to \mathbb{R}$ a quasi-convex function w.r.t. $(x, \xi, t) \in X \times \Xi \times T_{\max}$ that is upper semi-continuous w.r.t. $t \in T(x,\xi)$ for all fixed $(x,\xi) \in X \times \Xi$. Additionally, let T be a compact-valued correspondence satisfying $T(y(\lambda)) \subseteq \lambda T(y_1) + (1-\lambda)T(y_2)$ with $y(\lambda) = \lambda y_1 + (1-\lambda)y_2, y_1 = (x_1,\xi_1), y_2 = (x_2,\xi_2)$ for any $\lambda \in [0,1], x_1, x_2 \in X, \xi_1, \xi_2 \in \Xi$. Then the probust function φ is α-concave.

Proof. We show this claim using Corollary 1.2.21 and Theorem 1.2.20.
First we show that the function $g(x,\xi,T) := \max_{t \in T(x,\xi)} g(x,\xi,t)$ is a quasi-convex function.
For this, let us now fix any arbitrary $x_1, x_2 \in X$, arbitrary $\xi_1, \xi_2 \in \Xi$ and any $\lambda \in [0,1]$. Denote $y_1 = (x_1,\xi_1), y_2 = (x_2,\xi_2)$.
Since T satisfies the condition

$$T(y(\lambda)) \subseteq \lambda T(y_1) + (1-\lambda)T(y_2),$$

we know that for any $t \in T(y(\lambda))$ there exists $t_1 \in T(y_1), t_2 \in T(y_2)$ such that the scenario t can be represented as $t = \lambda t_1 + (1-\lambda)t_2$.
Consequently, due to the compactness of $T(x,\xi)$ for all $(x,\xi) \in X \times \Xi$ and the upper semi-continuity of $g(x,\xi,\cdot)$ we can choose any

$$t^* \in \arg \max_{t \in T(\lambda x_1 + (1-\lambda)x_2, \lambda\xi_1 + (1-\lambda)\xi_2)} g(\lambda x_1 + (1-\lambda)x_2, \lambda\xi_1 + (1-\lambda)\xi_2, t)$$

and estimate by the quasi-convexity of g

$$\begin{aligned}g(\lambda x_1+(1-\lambda)x_2,\lambda\xi_1+(1-\lambda)\xi_2,T) &= g(\lambda x_1+(1-\lambda)x_2,\lambda\xi_1+(1-\lambda)\xi_2,t^*)\\ &= g(\lambda x_1+(1-\lambda)x_2,\lambda\xi_1+(1-\lambda)\xi_2,\lambda t_1+(1-\lambda)t_2)\\ &\leq \max\{g(x_1,\xi_1,t_1),g(x_2,\xi_2,t_2)\}\\ &\leq \max\{g(x_1,\xi_1,T),g(x_2,\xi_2,T)\}.\end{aligned}$$

Because $x_1,x_2\in X,\xi_1,\xi_2\in\Xi,\lambda\in[0,1]$ were chosen arbitrarily, $g(x,\xi,T)$ is quasi-convex w.r.t. $(x,\xi)\in X\times\Xi$.
Now we use Corollary 1.2.21 to ensure that

$$\Gamma : X \rightrightarrows \mathbb{R}^m, x\mapsto \{\xi\in\Xi \mid \sup_{t\in T(x,\xi)} g(x,\xi,t)\leq 0\}$$

satisfies $\Gamma(x(\lambda))\supseteq\lambda\Gamma(x_1)+(1-\lambda)\Gamma(x_2)$ for all $x_1,x_2\in X,\lambda\in[0,1]$. Then we can use Theorem 1.2.20 which ensures that φ is a quasi-concave function w.r.t. $x\in X$. □

Generalized probust optimization problems

With continuity and concavity statements of probust functions additionally to the decision-dependent uncertainty concepts, semi-infinite optimization problems and chance constrained optimization problems, we are finally ready to tackle generalized probust optimization problems. We recall that a (standard) probust optimization problem has the form

$$\text{SPP} : \min_{x\in X} f(x) \text{ s.t. } \mathbb{P}(g(x,\xi,t)\leq 0\ \forall t\in T)\geq p,$$

where $X\subseteq\mathbb{R}^n$ is a compact set for $n\in\mathbb{N}$, $f:X\to\mathbb{R}$ is a continuous mapping, $\mathbb{P}$ is a probability measure on the measurable space $(\Xi,\mathcal{A})$, where $\Xi\subseteq\mathbb{R}^m, m\in\mathbb{N}$ and $\mathcal{A}$ is a σ-algebra on Ξ, $p\in[0,1], T\subseteq\mathbb{R}^q, q\in\mathbb{N}$ is a compact set and $g:X\times\Xi\times T\to\mathbb{R}$ is a continuous function w.r.t. (x,ξ,t).
There already exists a generalization by Adelhütte et al. [5] and van Ackooij et al. [3], where they assumed that the uncertainty set T depends on the decision x.

We generalize the definition of a probust optimization problem using the two decision-depending uncertainty concepts introduced in the last section:
First, the stochastics can be decision-dependent. This idea is studied so far, e.g., by Lejeune et al. [53] and Basciftci et al. [8] in a linear context. An overview about decision-dependent uncertainties can be found in a paper from Hellemo et al. [40].
Second, the set of scenarios is not constant. This idea is well-known from generalized semi-infinite optimization and studied for example by Still [81], Vázquez et al. [84] and Stein [80].
The resulting problem definition can be formulated as:

Definition 1.3.7 (Generalized probust optimization problem)
Let X be a set, $f : X \to \mathbb{R}$ be a function, $(\Xi, \mathcal{A})$ be a measurable space, $\mathbb{P} : X \to \mathcal{P}(\Xi, \mathcal{A})$ be a probability distribution function, $T : X \times \Xi \rightrightarrows \mathbb{R}^q$ be a correspondence and $g : X \times \Xi \times T_{\max} \to \mathbb{R}$ be a function that satisfies $\Omega(x) \in \mathcal{A}$ for all $x \in X$. Then we define the generalized probust optimization problem as

$$\text{GPP} : \min_{x \in X} f(x) \text{ s.t. } \mathbb{P}\,(x)\,(g(x, \xi, t) \leq 0 \;\forall t \in T(x, \xi)) \geq p.$$

Before we consider the solution existence theorem, we have to guarantee that the feasible set of the generalized probust optimization problem is not empty. We can check this by solving the following problem that is closely related to Equation (1.5):

$$p_{\max} = \max_{x \in X, p \in [0,1]} p \text{ s.t. } \mathbb{P}(x)(g(x, \xi, t) \leq 0 \;\forall t \in T(x, \xi)) \geq p \tag{1.6}$$

Lemma 1.3.8 (Well-definedness of $p_{\max}$)
Let the assumptions of Proposition 1.3.4 hold and $\emptyset \neq X \subseteq \mathbb{R}^n$ be a compact set. Then the value $p_{\max}$ in (1.6) is well-defined.

Proof. We show this claim directly by using Proposition 1.3.4 and Weierstrass' theorem: As all assumptions for Proposition 1.3.4 are fulfilled w.r.t. to $x \in X$, we know that the probust function $\varphi : X \to [0, 1]$ is upper semi-continuous. Consequently, the function

$$G(x, p) = p - \varphi(x)$$

is lower semi-continuous w.r.t. $(x, p) \in X \times [0, 1]$ as the difference of a continuous function and an upper semi-continuous function.
Therefore, the feasible set of problem (1.6) is compact as the intersection of the compact set $X \times [0, 1]$ with the closed set described by $G(x, p) \leq 0$.
Weierstrass' theorem then guarantees the existence of a maximum in the feasible set because the objective function is continuous w.r.t. $(x, p) \in X \times [0, 1]$.
Please note that the feasible set is not empty because X is not empty and thus every choice $(x, 0)$ with $x \in X$ is feasible. □

Using this additional information, we can ensure that the feasible set of a generalized probust optimization problem is not empty. This allows us to determine sufficient conditions to guarantee the existence of a solution of a generalized probust optimization problem:

Theorem 1.3.9 (Existence of a solution of GPP)
Let the assumptions of Lemma 1.3.8 hold. Additionally, let $f : X \to \mathbb{R}$ be a lower semi-continuous function and $p : X \to [0, 1]$ be a continuous threshold function such that there exists some $x \in X$ with $p(x) \leq p_{\max}$. Then the generalized probust optimization problem has a well-defined solution.

Proof. We show this claim directly by using Proposition 1.3.4 and a variant of Weierstrass' theorem:
As all assumptions for Proposition 1.3.4 are fulfilled w.r.t. to $x \in X$, we know that the probust function $\varphi : X \to [0,1]$ is upper semi-continuous. Consequently, the function

$$G(x) = p(x) - \varphi(x)$$

is lower semi-continuous w.r.t. $x \in X$ as the difference of a continuous function and an upper semi-continuous function.
Therefore, the feasible set of the generalized probust optimization problem is compact as the intersection of the compact set X with the closed feasible set.
A variant of Weierstrass' theorem then guarantees us the existence of a minimum of the objective function over the feasible set because f is lower semi-continuous w.r.t. $x \in X$.
Please note that the feasible set is not empty as by Lemma 1.3.8 the threshold $p_{\max}$ is well-defined and we assumed the existence of some $\overline{x} \in X$ such that $p(\overline{x}) \leq p_{\max}$. □

Next to the existence of a solution of a GPP, we are interested in the uniqueness of this solution. Since we have already found sufficient conditions for the concavity of a probust function, we can use these conditions to guarantee the uniqueness of a solution of a GPP:

Theorem 1.3.10 (Unique solution of GPP)
Let the assumptions of Theorem 1.3.6 and Theorem 1.3.9 hold. Furthermore, let $X \subseteq \mathbb{R}^n$ be a convex set, $f : X \to \mathbb{R}$ be strictly convex and $p : X \to [0,1]$ be a quasi-convex function. Then the generalized probust optimization problem has a unique minimizer x^*.

Proof. Due to the assumptions of Theorem 1.3.9 hold, we know that there exists some solution x^* of the generalized probust optimization problem.
We will show that this solution is unique by since the feasible set is convex and the objective function is strictly convex w.r.t. $x \in X$.
Because the assumptions of Theorem 1.3.6 are fulfilled, we know that φ is a quasi-concave function w.r.t. $x \in X$.
As p is quasi-convex, we know that $G(x) := p(x) - \varphi(x)$ is quasi-convex as the difference of a quasi-convex and a quasi-concave function. Since X is convex, the feasible set is convex as the intersection of two convex sets.
Because f was assumed to be strictly convex, we know that the minimizer of this problem is unique. □

In the other chapters of this thesis, we do not concentrate on generalized probust optimization problems, but on standard probust optimization problems instead. This reduction of problem structure does not necessarily change the solution of the problem as we show next.

Reduction of generalized probust optimization problems to standard probust optimization problems

In the last part of this section, we are interested in simplifying generalized probust optimization problems to standard probust optimization problems. One way to handle decision-dependent probability measures is the so called “push-in method” that can be found, e.g., in a note by Rubinstein et al. [68] and in a paper by Dupacova [29].
To apply this method, we assume that the decision-dependent probability measure can be represented by a probability density function $\rho : X \times \Xi \to \mathbb{R}$. This means that for all measurable sets $A \in \mathcal{A}$ we can rewrite the expected value of a random variable f as:

$$\begin{aligned}\mathbb{E}_x(f) &= \int_\Xi f(\xi) d\mathbb{P}_x(\xi)\\ &= \int_\Xi f(\xi)\rho(x,\xi)d\xi\\ &= \int_\Xi f(\xi)\frac{\rho(x,\xi)}{\hat{\rho}(\xi)}\hat{\rho}(\xi)d\xi\\ &= \int_\Xi f(\xi)\frac{\rho(x,\xi)}{\hat{\rho}(\xi)}d\hat{\mathbb{P}}(\xi)\\ &= \mathbb{E}\left(f\frac{\rho(x,\cdot)}{\hat{\rho}}\right),\end{aligned}$$

where the function $\hat{\rho} : \Xi \to \mathbb{R}$ is the probability density of a reference probability distribution $\hat{\mathbb{P}}$ on $(\Xi, \mathcal{A})$ which is independent of the a decision $x \in X$. Since this probability density $\hat{\rho}$ is pushed into the integrand, it gives the method its name.
Considering probust optimization problems, we are interested in $f = \chi_A$ for some measurable set $A \in \mathcal{A}$. We can reformulate probability evaluations by expected values via $\mathbb{P}_x(A) = \mathbb{E}_x(\chi_A)$ for any fixed $x \in X$, but unfortunately we cannot reformulate the last expression backwards into an indicator function as $\frac{\rho(x,\cdot)}{\hat{\rho}} \notin \{0,1\}$ in general.
Consequently, we have to use a new approach to simplify generalized probust optimization problems. We need a special structure of this problem to do so:

Assumption 1.3.11 (Existence of reference objects)
Given an instance of a generalized probust optimization problem GPP, we assume there exists a reference probability space $(\hat{\Xi}, \hat{\mathcal{A}}, \hat{\mathbb{P}})$ *and a reference set* $\hat{T} \subseteq \mathbb{R}^q$ *as well as two functions*

$$\begin{aligned}\mathcal{T}_{\hat{\Xi}} &: X \times \hat{\Xi} \to \Xi\\ \mathcal{T}_{\hat{T}} &: X \times \hat{\Xi} \times \hat{T} \to \mathbb{R}^q\end{aligned}$$

such that for every fixed $\overline{x} \in X$ *the function*

$$\mathcal{T}_{\hat{\Xi}}(\overline{x}, \cdot) : \hat{\Xi} \to \Xi$$

is a bijection between the spaces of realizations.

Furthermore, we assume that the function

$$\mathcal{T}_{\hat{\mathcal{A}}} : \hat{\mathcal{A}} \to \mathcal{A}, \hat{A} \mapsto \bigcup_{\omega \in \hat{A}} \{\mathcal{T}_{\hat{\Xi}}(\overline{x}, \omega)\}$$

is a bijective mapping between the σ-algebras satisfying

$$\hat{\mathbb{P}}(\hat{A}) = \mathbb{P}(\overline{x})(\mathcal{T}_{\hat{\mathcal{A}}}(\overline{x}, \hat{A}))\ \forall \hat{A} \in \hat{\mathcal{A}}$$

and for any fixed $(\overline{x}, \overline{\omega}) \in X \times \hat{\Xi}$, the function

$$\mathcal{T}_{\hat{T}}(\overline{x}, \overline{\omega}, \cdot) : \hat{T} \to T(\overline{x}, \mathcal{T}_{\hat{\Xi}}(\overline{x}, \overline{\omega}))$$

is a bijection between the two uncertainty sets.

Because these transformations are important for the application part of this thesis, we denote an extra definition for them:

Definition 1.3.12 (Transformation of uncertainty sets)
Given a generalized probust optimization problem GPP, a reference probability space $(\hat{\Xi}, \hat{\mathcal{A}}, \hat{\mathbb{P}})$, a reference set $\hat{T} \subseteq \mathbb{R}^q$ and two functions $\mathcal{T}_{\hat{\Xi}} : X \times \hat{\Xi} \to \Xi$, $\mathcal{T}_{\hat{T}} : X \times \hat{\Xi} \times \hat{T} \to \mathbb{R}^q$ that fulfill Assumption 1.3.11, then we call $(\hat{\Xi}, \hat{\mathcal{A}}, \hat{\mathbb{P}})$ a reference probability space of GPP, $\hat{T}$ a reference uncertainty set of GPP, $\mathcal{T}_{\hat{\Xi}}$ a realization transformation and $\mathcal{T}_{\hat{T}}$ a scenario transformation of the GPP.

Please note that these reference objects and transformation are not unique in general, because the reference sets and transformations are scalable.
With these transformations, we can reduce a generalized probust optimization problem to a standard probust optimization problem. The next theorem shows that transformations of uncertainty sets do not change the feasible set of the optimization problem.

Theorem 1.3.13 (GPP-reduction theorem)
Given a general probust optimization problem of the form

$$GPP : \min_{x \in X} f(x) \text{ s.t. } \mathbb{P}(x)\left(g(x, \xi, t) \leq 0\ \forall t \in T(x, \xi)\right) \geq p,$$

a reference probability space $(\hat{\Xi}, \hat{\mathcal{A}}, \hat{\mathbb{P}})$, a reference set $\hat{T} \subseteq \mathbb{R}^q$ and two transformations $\mathcal{T}_{\hat{\Xi}} : X \times \hat{\Xi} \to \Xi, \mathcal{T}_{\hat{T}} : X \times \hat{\Xi} \times \hat{T} \to \mathbb{R}^q$ that fulfill Assumption 1.3.11, then a decision is feasible for GPP if and only if this decision is feasible for the following standard probust optimization problem

$$SPP : \min_{x \in X} f(x)\ s.t.\ \hat{\mathbb{P}}(\tilde{g}(x, \omega, z) \leq 0\ \forall z \in \hat{T}) \geq p,$$

where

$$\tilde{g} : X \times \hat{\Xi} \times \hat{T} \to \mathbb{R}, (x, \omega, z) \mapsto g(x, \mathcal{T}_{\hat{\Xi}}(x, \omega), \mathcal{T}_{\hat{T}}(x, \mathcal{T}_{\hat{\Xi}}(x, \omega), z)).$$

Proof. We show this claim by proving one set inclusions, an indirect argument and the bijectivity of the transformations:
$\mathcal{F}_{\text{SPP}} \subseteq \mathcal{F}_{\text{GPP}}$: Let us fix an arbitrary $\overline{x} \in \mathcal{F}_{\text{SPP}}$, then we know by definition that

$$\hat{\mathbb{P}}(\tilde{g}(\overline{x}, \omega, z) \leq 0 \ \forall z \in \hat{T}) \geq p.$$

This means that there exists a set $\hat{A} \in \hat{\mathcal{A}}$ such that for all $\omega \in \hat{A}$ and all $z \in \hat{T}$ the inequality $\tilde{g}(\overline{x}, \omega, z) \leq 0$ is fulfilled and $\hat{\mathbb{P}}(\hat{A}) \geq p$.
To show $\overline{x} \in \mathcal{F}_{\text{GPP}}$, we have to construct a set $A \in \mathcal{A}$ such that $\mathbb{P}(\overline{x})(A) \geq p$ and for all $\xi \in A$ and all $t \in T(\overline{x}, \xi)$ the inequality $g(x, \xi, t) \leq 0$ is fulfilled.
Therefore, we fix the set $A := \mathcal{T}_{\hat{\mathcal{A}}}(\overline{x}, \hat{A})$. By Assumption 1.3.11 we know that $A \in \mathcal{A}$ and $\mathbb{P}(\overline{x})(A) = \hat{\mathbb{P}}(\hat{A}) \geq p$.
Now, we argue indirectly by assuming that there exists a realization $\xi \in A$ and a scenario $t \in T(\overline{x}, \xi)$ such that $g(\overline{x}, \xi, t) > 0$.
By the definition of $\mathcal{T}_{\hat{\mathcal{A}}}$ and the bijectivity of $\mathcal{T}_{\hat{\Xi}}(\overline{x}, \cdot)$, we know there exists a $\overline{\omega} \in \hat{A}$ such that $\mathcal{T}_{\hat{\Xi}}(\overline{x}, \overline{\omega}) = \xi$ and by the bijectivity of $\mathcal{T}_{\hat{T}}(\overline{x}, \overline{\omega}, \cdot)$, we know that there exists a $\overline{z} \in \hat{T}$ such that $\mathcal{T}_{\hat{T}}(\overline{x}, \overline{\omega}, \overline{z}) = t$.
Consequently, we know that

$$0 < g(\overline{x}, \xi, t) = g(\overline{x}, \mathcal{T}_{\hat{\Xi}}(\overline{x}, \overline{\omega}), \mathcal{T}_{\hat{T}}(\overline{x}, \mathcal{T}_{\hat{\Xi}}(\overline{x}, \overline{\omega}), \overline{z})) = \tilde{g}(\overline{x}, \overline{\omega}, \overline{z}).$$

This contradicts $\overline{\omega} \in \hat{A}$ and therefore the basic assumption which implies that $\overline{x} \in \mathcal{F}_{\text{GPP}}$.
Because $\overline{x} \in \mathcal{F}_{\text{SPP}}$ was chosen arbitrarily, the set inclusion is guaranteed.
Due to the bijectivity of the transformations, the other set inclusion is also satisfied. Altogether the claim holds. □

Please note, that we chose the same objective function for both optimization problems GPP and SPP. Therefore and because of the invariance of the feasible sets, we know that global (or local) optima of GPP are global (or local) optima of SPP and vice versa.

While transformation examples from a decision depending set to a reference set can be found in a paper of Schwientek et al. [71], the transformation to a reference probability space is new concept to the best of our knowledge. While the push-in-method from Rubinstein's paper [52] is roughly speaking a multiplication of the integrand 1 our transformation approach changes the representation of the uncertain parameters.
We can understand this change of representation better considering some examples:

Example 1.3.14 (Extended Example 1.2.14)
(*i*) *Transform the realizations of the decision-dependent random variable* $Z \sim \mathcal{U}([a(x), b(x)])$ *by the (inverse) mapping*

$$\mathcal{T}_{\hat{\Xi}}^{-1} : X \times \Xi, (x, \xi) \mapsto \omega := \frac{\xi - a(x)}{b(x) - a(x)}.$$

This implies a reference random variable $\hat{Z} \sim \mathcal{U}([0,1])$ and we can confirm for an arbitrary measurable set $A \in \mathcal{B}$

$$\begin{aligned}
\mathbb{P}(x)(A) &= \int_{\xi \in A} 1 d\mathbb{P}(x)(\xi) \\
&= \int_{\xi \in A \cap [a(x), b(x)]} \frac{1}{b(x)-a(x)} d\lambda(\xi) \\
&= \int_{\omega \in \mathcal{T}_{\hat{\Xi}}(B \cap [a(x), b(x)])} 1 d\lambda(\omega) \\
&= \int_{\omega \in \mathcal{T}_{\hat{\Xi}}(A) \cap [0,1]} 1 d\lambda(\omega) \\
&= \int_{\omega \in \hat{A}} 1 d\tilde{\mathbb{P}}(\omega) \\
&= \hat{\mathbb{P}}(\hat{A}),
\end{aligned}$$

where $\hat{A} = \left\{ \hat{\xi} \in \mathbb{R} \mid \exists \xi \in \Xi : \hat{\xi} = \frac{\xi - a(x)}{b(x)-a(x)} \right\} = \frac{A-a(x)}{b(x)-a(x)}$.
Reformulating the last equality leads to the transformation of measurable sets

$$\mathcal{T}_{\hat{\mathcal{A}}} : X \times \mathcal{B} \to \mathcal{B}, (x, \hat{A}) \mapsto (b(x)-a(x))\hat{A} + a(x).$$

(ii) *Transform the realizations of the decision-dependent random variable $Z \sim \mathcal{N}(\mu(x), \sigma(x))$ by the (inverse) mapping*

$$\mathcal{T}_{\hat{\Xi}}^{-1} : X \times \Xi, (x, \xi) \mapsto \omega := \frac{\xi - \mu(x)}{\sigma(x)}.$$

This implies a reference random variable $\hat{Z} \sim \mathcal{N}(0,1)$ and we can confirm for an arbitrary measurable set $A \in \mathcal{B}$

$$\begin{aligned}
\mathbb{P}(x)(A) &= \int_{\xi \in A} 1 d\mathbb{P}(x)(\xi) \\
&= \int_{\xi \in A} \frac{1}{\sqrt{2\pi}\sigma(x)} \exp\left(-\frac{(\xi - \mu(x))^2}{2\sigma(x)^2} \right) d\lambda(\xi) \\
&= \int_{\omega \in \mathcal{T}_{\hat{\Xi}}(A)} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\omega^2}{2} \right) d\lambda(\omega) \\
&= \int_{\omega \in \hat{A}} 1 d\tilde{\mathbb{P}}(\omega) \\
&= \hat{\mathbb{P}}(\hat{A}),
\end{aligned}$$

where $\hat{A} = \left\{ \hat{\xi} \in \mathbb{R} \mid \exists \xi \in \Xi : \hat{\xi} = \frac{\xi - \mu(x)}{\sigma(x)} \right\} = \frac{A - \mu(x)}{\sigma(x)}$.
Reformulating the last equality leads to the transformation of measurable sets

$$\mathcal{T}_{\hat{\mathcal{A}}} : X \times \mathcal{B} \to \mathcal{B}, (x, \hat{A}) \mapsto \sigma(x)\hat{A} + \mu(x).$$

Here the push-in technique would define a second factor within the integrand which the transformation approach does not.
Since we have shown in the proof of Theorem 1.3.13 that the probust functions of GPP and the corresponding SPP calculate the same probability for all $x \in X$, it does not change under the realization and scenario transformation. We state this as a corollary, but skip its proof as it is induced by the proof of the last theorem.

Corollary 1.3.15 (Invariance of sets of feasible realizations)
Given a generalized probust optimization problem GPP, a reference set $\hat{T} \subseteq \mathbb{R}^q$ and a function $\mathcal{T}_{\hat{T}} : X \times \Xi \times \hat{T} \to \mathbb{R}$ that is a bijection between $\hat{T}$ and $T(\overline{x}, \overline{\xi})$ for all fixed $(\overline{x}, \overline{\xi}) \in X \times \Xi$. Then for any fixed $\overline{x} \in X$ the set of feasible realizations is invariant under this transformation, meaning

$$\begin{aligned}\Omega(\overline{x}) &= \{\xi \in \Xi \mid g(\overline{x}, \xi, t) \leq 0 \ \forall t \in T(x, \xi)\} \\ &= \{\xi \in \Xi \mid \tilde{g}(\overline{x}, \xi, z) \leq 0 \ \forall z \in \hat{T}\},\end{aligned}$$

where $\tilde{g}(x, \xi, z) := g(x, \xi, \mathcal{T}_{\hat{T}}(x, \xi, z))$ for all $(x, \xi, z) \in X \times \Xi \times \hat{T}$.

Although the probust function is not influenced by the transformations, the inner functions g and $\tilde{g}$ change with these transformations. If we assume that the transformations are at least continuous, we know that continuity of g can be transmitted to continuity of $\tilde{g}$ as a composition of continuous functions. On the contrary, we have seen in Example 7.3.1 in [72] that convexity has not to be transmitted from g to $\tilde{g}$. A similar behavior is mentioned in Dupacova's work [28], where the push-in technique can destroy concavity in the probability evaluation.
As the statements of Propositions 1.1.6 - 1.1.8 from (standard) probust optimization in Section 1.1 showed: Preserving continuity of the inner function leads to preserving the existence of a solution of a probust optimization problem, while loosing convexity can lead to the loss of a unique solution. The question how we can find a solution of a probust optimization problem numerically will be answered in the next part of this thesis.

Overview of Chapter 1

In Chapter 1 we have introduced the concept of probust optimization problems. Because this first chapter handles a lot of concepts, we want to visualize the main points of this chapter. We focus on the concepts that form (standard) probust optimization problems in Figure 1.4. Furthermore, we summarize the workflow of how to handle generalized probust optimization problems so far in Figure 1.5.

We have classified the already known concept of (standard) probust optimization problems by considering certain subproblems of this problem class. In this context, we fixed either a decision, a realization of the random variable or a scenario of the uncertainty set.

We have gathered results according to these subproblems that are known in the literature and guarantee the (unique) existence of a solution. We can now express the relations between the subproblems and the (standard) probust optimization problem visually by the following figure:

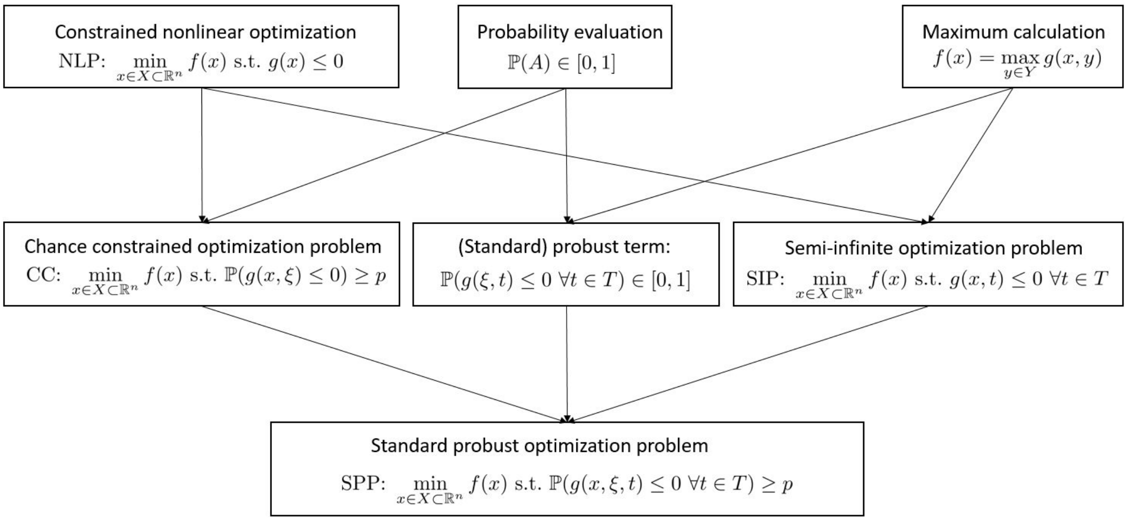

Constrained nonlinear optimization
NLP: $\min_{x\in X\subset\mathbb{R}^n} f(x)$ s.t. $g(x)\le 0$
Probability evaluation
$\mathbb{P}(A)\in[0,1]$
Maximum calculation
$f(x)=\max_{y\in Y} g(x,y)$
Chance constrained optimization problem
CC: $\min_{x\in X\subset\mathbb{R}^n} f(x)$ s.t. $\mathbb{P}(g(x,\xi)\le 0)\ge p$
(Standard) probust term:
$\mathbb{P}(g(\xi,t)\le 0\ \forall t\in T)\in[0,1]$
Semi-infinite optimization problem
SIP: $\min_{x\in X\subset\mathbb{R}^n} f(x)$ s.t. $g(x,t)\le 0\ \forall t\in T$
Standard probust optimization problem
SPP: $\min_{x\in X\subset\mathbb{R}^n} f(x)$ s.t. $\mathbb{P}(g(x,\xi,t)\le 0\ \forall t\in T)\ge p$

Figure 1.4: Decomposition of probust-optimization problems

As we found nothing on probust terms in the literature, we filled this gap in Section 1.3 and generalized the concept of probust optimization problems. Furthermore, we guaranteed that these problems have a (unique) solution using results from decision-dependent uncertainty concepts and correspondences.
Last, but not least, we introduced transformations that allow us to reduce a generalized probust optimization problem to a standard probust optimization problem.

Changing our perspective from modelling to solving optimization problems, we have to ask ourselves how we can handle the introduced problem classes. So far, we know that we can reduce a generalized probust optimization problem to a standard probust optimization problem using appropriate transformations. If we can express the worst-case scenarios analytically, we can reduce the problem even further to a (joint) chance constrained optimization problem that can be handled approximately or even analytically if we know enough about the set of feasible realizations. This workflow of handling generalized probust optimization problems is visualized by Figure 1.5.
Because no algorithm is known so far that handles probust optimization problems which cannot be reduced to joint chance constrained optimization problems someone has to solve the specific problem instances by hand.
The next part of this thesis handles numerical solving strategies that work directly with a standard probust optimization problem and do not need to express all worst-case scenarios analytically.

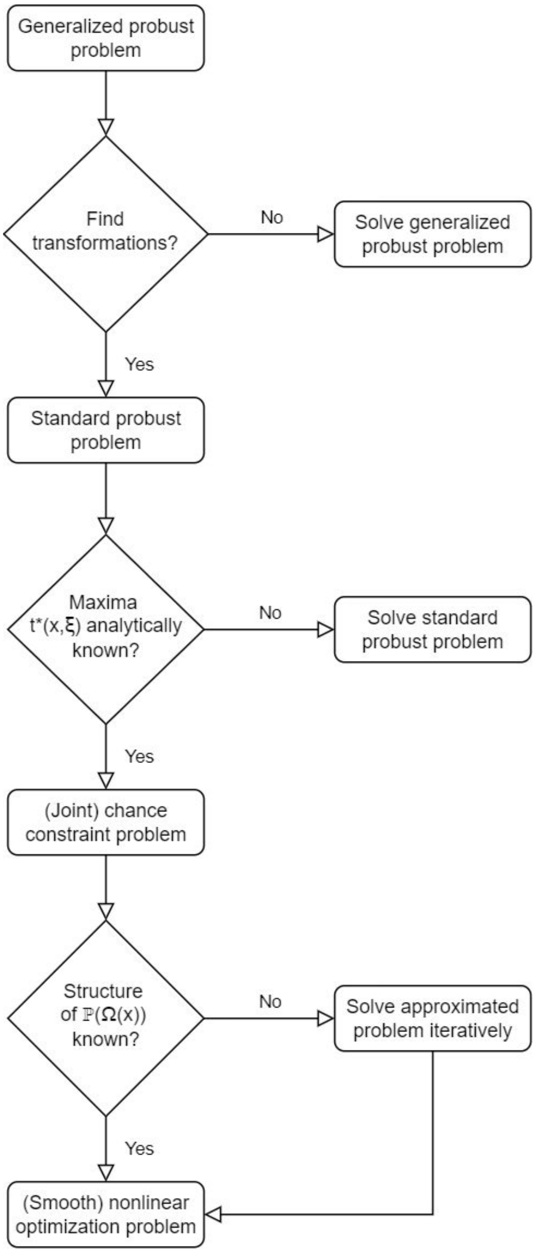

Generalized probust problem
Find transformations?
No
Solve generalized probust problem
Yes
Standard probust problem
Maxima t*(x,ξ) analytically known?
No
Solve standard probust problem
Yes
(Joint) chance constraint problem
Structure of ℙ(Ω(x)) known?
No
Solve approximated problem iteratively
Yes
(Smooth) nonlinear optimization problem

Figure 1.5: Workflow of handling generalized probust optimization problems

2 Probust subset schemes

After introducing probust optimization problems in the last chapter and giving sufficient conditions to guarantee the existence of an (unique) minimizer, we are interested in solving such problems.

We have already discussed one way to solve such problems analytically in Example 1.1.9. As the single steps of this process are not always analytically solvable, we ask ourselves how numerical methods can bypass these difficulties.

The main goal of this chapter is to solve probust optimization problems via discretization methods. We are inspired by the discretization schemes from semi-infinite optimization. As mentioned in the introduction, we can represent the uncertainty set T in semi-infinite optimization problems by one (decision-dependent) worst-case scenario. Consequently, we are looking for one scenario to represent the semi-infinite constraint.

In probust optimization this worst-case scenario does also depend on the realization of the random vector such that we have to consider a family of worst-case scenarios.

To be able to understand if and when a subset $S \subseteq T$ represents this family of worst-case scenarios, we need a deeper insight into the structure of a probust optimization problem. We start with focusing on probust terms where the decision $x \in X$ is fixed. We then transfer this insight to probust functions and study their convergence, when considering a subset sequence $(T_k)_{k\in\mathbb{N}}, T_k \subseteq T$ for all $k \in \mathbb{N}$.

With this knowledge, we can not only state the iterative probust subset algorithm to solve probust optimization problems approximately, but we can also show its convergence for quite general assumptions concerning the subset sequence $(T_k)_{k\in\mathbb{N}}$.

We close this chapter by giving example discretization schemes Φ, whose iterates converge towards the minimizer of the corresponding standard probust optimization problem. These examples include a uniform discretization approach as well as two probust versions of the adaptive discretization approach from Blankenship and Falk. These schemes are used as the starting point in Part II of this thesis to design specified discretization schemes to solve the given applications efficiently.

2.1 Discretized probust terms and functions

We introduced probust terms and probust functions as a subproblem of probust optimization problems in Section 1.3.

In this section we focus on the behaviour of the mapping $\varphi : 2^T \to [0, 1]$, where 2^T denotes the power set of T.

To guarantee that all terms $\varphi(S)$ with $S \in 2^T$ are well-defined, we make the following assumption:

Assumption 2.1.1 (Well-posedness of $\varphi(S)$)
Given a probust term $\mathbb{P}(g(\xi,t) \leq 0\ \forall t \in T)$, let $\Omega(S) := \{\xi \in \Xi \mid g(\xi,t) \leq 0\ \forall t \in S\} \in \mathcal{A}$ hold for all $S \in 2^T$.

Recalling Proposition 1.3.2, we already have sufficient conditions that ensure $\Omega \in \mathcal{A}$ and also $\Omega(S) \in \mathcal{A}$ for all $S \subseteq T$. Consequently, Assumption 2.1.1 is fulfilled for all $S \subseteq T$, if g is lower semi-continuous w.r.t. $(\xi,t) \in \Xi \times T$ and if we consider a Borel-σ-algebra.
After this comment on the well-posedness of φ, we show that φ is monotone decreasing for increasing subsets w.r.t. the partial order $(\subseteq, 2^T)$.

Proposition 2.1.2 (Monotonicity of φ w.r.t. $\subseteq$)
Given a probability space $(\Xi,\mathcal{A},\mathbb{P})$, any set T, two subsets $S_1 \subseteq S_2 \subseteq T$ and an inner function $g : \Xi \times T \to \mathbb{R}$. Assume that $\Omega(S_1), \Omega(S_2) \in \mathcal{A}$, then

$$\varphi(S_2) \leq \varphi(S_1).$$

Proof. We show the claim directly. Let us fix any set T and two subsets $S_1 \subseteq S_2 \subseteq T$. Due to $S_1 \subseteq S_2$, we know $\bigcap_{t\in S_2} \Omega(t) \subseteq \bigcap_{t\in S_1} \Omega(t)$ and by the monotonicity of probability measures we can write:

$$\begin{aligned}\varphi(S_2) &= \mathbb{P}\left(\bigcap_{t\in S_2} \Omega(t)\right)\\ &\leq \mathbb{P}\left(\bigcap_{t\in S_1} \Omega(t)\right)\\ &= \varphi(S_1)\end{aligned}$$

Because we fixed the subsets $S_1, S_2 \subseteq T$ arbitrarily, the claim holds. □

Consequently, an increasing set of discretization points will lead to a more and more precise approximation of φ. We are interested in subsets $S \subseteq T$ such that S is as small as possible to minimize computational efforts later on in Part II of this thesis, but also guarantees $\varphi(S) = \varphi(T)$.
Checking the size of S is manageable, but the second condition $\varphi(S) = \varphi(T)$ has to be quantifiable. Therefore, we introduce the following condition:

Definition 2.1.3 (Bottle-neck-condition)
Given a probust term $\varphi = \mathbb{P}(g(\xi,t) \leq 0\ \forall t \in T)$ such that Assumption 2.1.1 *is satisfied, we say that $S \subseteq T$ satisfies the bottle-neck-condition at $\xi \in \Xi$ if:*

$$\forall t \in T\ \exists s = s(\xi,t) \in S : g(\xi,s) \leq 0 \Rightarrow g(\xi,t) \leq 0$$

We say that a subset $S \subseteq T$ satisfies the bottle-neck-condition if there exists some measurable set $M \in \mathcal{A}$ such that

$$\mathbb{P}(M) = 1 \text{ and } S \text{ fulfills the bottle-neck-condition for all } \xi \in M.$$

We say $\emptyset$ satisfies the bottle-neck-condition if $\varphi(T) = 1$.

To understand this new definition better, we investigate the following example:

Example 2.1.4 (Examples of subsets fulfilling the bottle-neck-condition)
Consider the probability space $(\mathbb{R}, \mathcal{B}, \mathbb{P})$, where $\mathcal{B}$ is the Borel-σ-algebra on $\mathbb{R}$ and $\mathbb{P}$ is the probability measure induced by $Z \sim \mathcal{U}([0,1])$ and $T := [0,1]$.
Let two different inner functions $g_1, g_2 : \Xi \times T \to \mathbb{R}$ be defined by

$$\begin{aligned} g_1(\xi, t) &:= \xi - t^2 \text{ and} \\ g_2(\xi, t) &:= \xi - (\xi - t)^2. \end{aligned}$$

Then any set $S \supseteq \{0\}$ satisfies the bottle-neck-condition regarding the inner function g_1 since any fixed $\xi \in \Xi$ with $g(\xi, 0) \leq 0$ implies $g(\xi, t) = \xi - t^2 \leq 0$ for all $t \in [0,1]$.
For the second inner function g_2 the bottle-neck-condition holds only for sets S that are the same as T up to a $\mathbb{P}$-null set such as $S = T$. Of course this choice is trivial to satisfy the bottle-neck-condition.

If the inner function g is continuous w.r.t. $(\xi, t) \in \Xi \times T$ jointly and $T \subseteq \mathbb{R}^q$ is compact, then the functions

$$\begin{aligned} g(\cdot, S) &: \Xi \to \mathbb{R}, \xi \mapsto \sup_{s \in S} g(\xi, s) \text{ and} \\ g(\cdot, T) &: \Xi \to \mathbb{R}, \xi \mapsto \max_{t \in T} g(\xi, t) \end{aligned}$$

are continuous w.r.t. ξ due to Berge's maximum theorem 1.2.8.
Consequently, this implies for any subset $S \subseteq T$ that

$$\begin{aligned} M &:= \{\xi \in \Xi \mid S \text{ fulfills the bottle-neck-condition at } \xi\} \\ &= \{\xi \in \Xi \mid g(\xi, S) \leq 0 \Rightarrow g(\xi, T) \leq 0\} \\ &= \{\xi \in \Xi \mid g(\xi, S) \leq 0 \Leftrightarrow g(\xi, T) \leq 0\} \\ &= \{\xi \in \Xi \mid (g(\xi, S) \leq 0 \wedge g(\xi, T) \leq 0) \vee (g(\xi, S) > 0 \wedge g(\xi, T) > 0)\} \\ &= (\Omega(S) \cap \Omega(T)) \cup (\Omega^C(S) \cap \Omega^C(T)) \end{aligned}$$

is measurable as a union of finite intersections of measurable sets. Ensuring that Assumption 2.1.1 holds, this means M is measurable and therefore the bottle-neck-condition is well-defined.
That the bottle-neck-condition is sufficient to guarantee $\varphi(S) = \varphi(T)$ is the statement of the following proposition:

Proposition 2.1.5 (Characterization bottle-neck-condition)
Given a probability space $(\Xi, \mathcal{A}, \mathbb{P})$ and a continuous inner function $g : \Xi \times T \to \mathbb{R}$ w.r.t. $t \in T$ satisfying Assumption 2.1.1. Then a subset $S \subseteq T$ of a compact set $T \subseteq \mathbb{R}^q$ satisfies the bottle-neck-condition if and only if $\varphi(S) = \varphi(T)$.

Proof. We show the claim by proving both implications directly.
⇒: Let us assume that $S \subseteq T$ fulfills the bottle-neck-condition. Then there exists a set $M \in \mathcal{A}$ such that

$$\begin{aligned}
&\mathbb{P}(M^C) = 0 \text{ and} \\
&\forall \xi \in M, t \in T \; \exists s(\xi, t) \in S : g(\xi, s) \leq 0 \Rightarrow g(\xi, t) \leq 0
\end{aligned}$$

Fixing any $\xi \in M$ and taking the supremum over $t \in T$, we can rewrite this as

$$\sup_{t \in T} g(\xi, s(\xi, t)) \leq 0 \Rightarrow \sup_{t \in T} g(\xi, t) \leq 0.$$

As $s(\xi, t) \in S$ for all $t \in T$ and T is compact, we can rewrite

$$\max_{s \in \mathrm{cl}(S)} g(\xi, s) \leq 0 \Rightarrow \max_{t \in T} g(\xi, t) \leq 0.$$

This implies

$$\begin{aligned}
&\{\xi \in M \mid \max_{s \in \mathrm{cl}(S)} g(\xi, s) \leq 0\} \subseteq \{\xi \in M \mid \max_{t \in T} g(\xi, t) \leq 0\} \text{ and} \\
&\mathbb{P}(\max_{s \in \mathrm{cl}(S)} g(\xi, t) \leq 0 \mid \xi \in M) \leq \mathbb{P}(\max_{t \in T} g(\xi, t) \leq 0 \mid \xi \in M).
\end{aligned}$$

Consequently due to $S \subseteq T$, Proposition 2.1.2 and $\mathbb{P}(M^C) = 0$:

$$\begin{aligned}
\varphi(T) &\leq \varphi(S) \\
&= \mathbb{P}(g(\xi, s) \leq 0 \text{ for all } s \in S) \\
&= \mathbb{P}(\max_{s \in \mathrm{cl}(S)} g(\xi, s) \leq 0) \\
&= \mathbb{P}(\max_{s \in \mathrm{cl}(S)} g(\xi, s) \leq 0 \mid \xi \in M) + \mathbb{P}(\max_{s \in \mathrm{cl}(S)} g(\xi, s) \leq 0 \mid \xi \notin M) \\
&\leq \mathbb{P}(\max_{t \in T} g(\xi, t) \leq 0 \mid \xi \in M) + \mathbb{P}(\max_{s \in \overline{S}} g(\xi, s) \leq 0 \mid \xi \notin M) \\
&= \mathbb{P}(\max_{t \in T} g(\xi, t) \leq 0 \mid \xi \in M) + 0 \\
&= \mathbb{P}(\max_{t \in T} g(\xi, t) \leq 0) \\
&= \varphi(T)
\end{aligned}$$

This proofs the first implication.

⇐: Let us assume that $\varphi(S) = \varphi(T)$ holds. By definition of φ, we can write:

$$\begin{aligned}
&\varphi(S) = \varphi(T) \\
\Leftrightarrow\; &\mathbb{P}(\Omega(S)) - \mathbb{P}(\Omega(T)) = 0
\end{aligned}$$

Because $S \subseteq T$ and by the definition of $\Omega(S)$, we know that $\Omega(T) \subseteq \Omega(S)$ and therefore

$$0 = \mathbb{P}(\Omega(S)) - \mathbb{P}(\Omega(T)) = \mathbb{P}(\Omega(S)/\Omega(T)).$$

This means that the set

$$\begin{aligned} N &:= \Omega(S)/\Omega(T) \\ &= \{\xi \in \Xi \mid g(\xi, s) \leq 0 \ \forall s \in S \ \wedge \ \max_{t \in T} g(\xi, t) > 0\} \\ &= \{\xi \in \Xi \mid g(\xi, s) \leq 0 \ \forall s \in S \ \wedge \ \exists t \in T : g(\xi, t) > 0\} \end{aligned}$$

is a $\mathbb{P}$-null set.
Because N defines the points where the bottle-neck-condition does not hold, we can conclude that S fulfills the bottle-neck-condition. This shows the second implication.
All together the claim holds. □

Next, we talk about continuity properties of φ as we assume an additional condition to hold. It is motivated by the results of Farshbaf-Shaker et al. (see Proposition 1.1.7):

Assumption 2.1.6 (Assumptions concerning $(\Xi, \mathcal{A}, \mathbb{P})$)
Assume that Assumption 2.1.1 *holds. Assume further that the considered probability distribution has a Lebesgue-density, meaning that there exists a function $\rho : \Xi \to \mathbb{R}_{\geq 0}$ such that we can calculate the probability of any measurable set $A \in \mathcal{A}$ by*

$$\mathbb{P}(A) = \int_{\xi \in A} \rho(\xi) d\lambda(\xi),$$

where λ is the Lebesgue-measure on Ξ.
Additionally assume that given a probust term φ and any $S \subseteq T$, it holds

$$\mathbb{P}\left(\max_{t \in \mathrm{cl}(S)} g(\xi, t) = 0\right) = 0. \tag{2.1}$$

Please note that the statement $\mathbb{P}(g(\xi, t) = 0) = 0$ for all $t \in T$ does not guarantee Equation 2.1 to hold as the following counter example shows:

Example 2.1.7 (Counter example for scenario-wise probability assumption)
Consider $g(\xi, t) := -(\xi - t)^2$, where the probability distribution is implied by the random variable $Z \sim \mathcal{U}([0, 1])$ and $t \in T := [0, 1]$.
Then we know on the one hand that

$$\mathbb{P}(g(\xi, t) = 0) = \mathbb{P}(\xi = t) = 0$$

for each fixed scenario $t \in T$.
But on the other hand, we know also that

$$\mathbb{P}(\max_{t \in T} g(\xi, t) = 0) = \mathbb{P}(0 = 0) = 1.$$

Consequently, the element-wise equation does not imply the set-wise equation.

We use this strengthened setting to show that adding a new point $t \in T$ to a given subset $S \subseteq T$ is a continuous function:

Proposition 2.1.8 (Adding one discretization point is a continuous operation)
Given a probability space $(\Xi, \mathcal{B}, \mathbb{P})$, a continuous inner function $g : \Xi \times T \to \mathbb{R}$ w.r.t. $(\xi, t) \in \Xi \times T$ that satisfies Assumption 2.1.6, a compact set $\emptyset \neq T \subseteq \mathbb{R}^q$ and a set $S \subseteq T$. Then the function

$$f_S : T \to [0,1], t \to \varphi(S \cup \{t\})$$

is continuous w.r.t. $t \in T$.

Proof. We show this claim directly by Theorem 1.1.3:
Let us fix a bounded set $\emptyset \neq T \subset \mathbb{R}^q$ and $S \subseteq T$. Because of the Berge's maximum theorem 1.2.8 and the continuity of g w.r.t. $(\xi, t) \in \Xi \times T$, we can conclude that the function $g(\cdot, S \cup \{t\}) := \sup_{s \in S \cup \{t\}} g(\cdot, s)$ is continuous w.r.t. $\xi \in \Xi$.
Due to Assumption 2.1.6 and by Theorem 1.1.3 the function $\varphi(S \cup \{\cdot\})$ is continuous. □

By adding single discretization points to a fixed set $S \subseteq T$, we are now interested in an iterative procedure creating a sequence of subsets $(T_k)_{k \in \mathbb{N}}, T_k \subseteq T, k \in \mathbb{N}$.
To handle an iterative procedures we have to define a stopping criterion that can be checked by the information at hand in a fixed iteration.
One option for such a stopping criterion is the following:

Definition 2.1.9 (Candidate-condition)
We say that for a probust term φ the candidate-condition holds for a subset $S \subseteq T$, if

$$\forall t \in T, \epsilon > 0 : \varphi(S) - \varphi(S \cup \{t\}) < \epsilon.$$

This criterion concentrates on the change of the probability $\varphi(S)$ instead on the structure of S itself like the bottle-neck-condition does.
Although the representation of these conditions are quite different, they share the following relation:

Theorem 2.1.10 (Candidate-condition and bottle-neck-condition)
Let the inner function g be continuous w.r.t. $(\xi, t) \in \Xi \times T$ and let Assumption 2.1.6 hold. Then a subset $S \subseteq T$ satisfies the bottle-neck-condition if and only if it fulfills the candidate-condition.
Furthermore, a subset S fulfilling one of these conditions implies

$$\varphi(S) = \varphi(T).$$

Proof. We prove the claim by showing both implications:
⇒: Let us assume that the candidate-condition holds.
We argue indirectly by assuming that the bottle-neck-condition is not fulfilled and can formalize

$$\begin{aligned} &\forall t \in T, \epsilon > 0 : \varphi(S) - \varphi(S \cup \{t\}) < \epsilon \text{ and} \\ &\exists \Omega \in \mathcal{A} : \mathbb{P}(\Omega) > 0 \text{ and} \\ &\qquad \forall \xi \in \Omega \exists \bar{t} \in T\ \forall s \in S : g(\xi, s) \leq 0 \text{ and } g(\xi, \bar{t}) > 0. \end{aligned} \tag{2.2}$$

Because g is continuous w.r.t. $(\xi, t) \in \Xi \times T$ and T is compact, we know using Berge's maximum theorem 1.2.8 that

$$g(\xi, S) := \sup_{s \in S} g(\xi, s)$$

is continuous w.r.t. $\xi \in \Xi$ and we can rewrite (2.2) as

$$\begin{aligned} &\forall \xi \in \Omega : g(\xi, S) \leq 0 \text{ and} \\ &\forall \xi \in \Omega \exists t \in T : g(\xi, t) > 0. \end{aligned}$$

Because we assumed $\mathbb{P}(\max_{s \in \mathrm{cl}(S)} g(\xi, t) = 0) = 0$ for all $S \subseteq T$, we can find a null-set N and some measurable set $\tilde{\Omega} \in \mathcal{A}$ such that

$$\begin{aligned} &\tilde{\Omega} = \Omega / N, \\ &\forall \xi \in \tilde{\Omega} : g(\xi, S) < 0 \text{ and} \\ &\forall \xi \in \tilde{\Omega} \exists t \in T : g(\xi, t) > 0. \end{aligned}$$

Due to the definition of N and $\tilde{\Omega}$, we know that $\mathbb{P}(\tilde{\Omega}) = \mathbb{P}(\Omega) > 0$ and therefore $\tilde{\Omega} \neq \emptyset$.
Now let us fix an arbitrary $\bar{\xi} \in \tilde{\Omega} \neq \emptyset$.
By definition of $\tilde{\Omega}$, we can find a $\bar{t} \in T$, such that $g(\bar{\xi}, \bar{t}) > 0$.
Due to the continuity of $g(\cdot, S)$ and $g(\cdot, \bar{t})$ w.r.t. $\xi \in \Xi$ we can find some $r > 0$ such that

$$\begin{aligned} &g(\xi, \bar{t}) > 0 \; \forall \xi \in B_r(\bar{\xi}) \text{ and} \\ &g(\xi, S) < 0 \; \forall \xi \in B_r(\bar{\xi}). \end{aligned}$$

Due to $\bar{\xi} \in \mathrm{supp}(\mathbb{P}) = \{\xi \in \Xi \mid \forall r > 0 : \mathbb{P}(B_r(\xi)) > 0\}$ and the definition of $g(\cdot, S)$ we can conclude

$$\varphi(S) - \varphi(S \cup \{\bar{t}\}) \quad \geq \quad \mathbb{P}(B_r(\bar{\xi})) > 0.$$

This contradicts the candidate-condition. Therefore our basic assumption (2.2) is wrong and we can assure that the bottle-neck-condition holds.

$\Leftarrow$: Let us now assume that the bottle-neck-condition holds. By Theorem 2.1.5 we know that $\varphi(S) = \varphi(T)$ holds. Because $S \subseteq T$ and the definition of $\Omega(S)$, we know that $\Omega(T) \subseteq \Omega(S)$ and therefore

$$0 = \mathbb{P}(\Omega(S)) - \mathbb{P}(\Omega(T)) = \mathbb{P}(\Omega(S)/\Omega(T)).$$

This means that the set $N := \Omega(S)/\Omega(T)$ is a $\mathbb{P}$ null-set.
Now we fix any arbitrary $\bar{t} \in T$ and because of $\Omega(T) \subseteq \Omega(S \cup \{\bar{t}\})$ we know

$$\Omega(S)/\Omega(S \cup \{\bar{t}\}) \subseteq \Omega(S)/\Omega(T).$$

Therefore $\Omega(S)/\Omega(S \cup \{\bar{t}\})$ is also a $\mathbb{P}$ null-set and consequently:

$$\begin{aligned} &\mathbb{P}(\Omega(S)/\Omega(S \cup \{\bar{t}\})) = 0 \\ \Leftrightarrow \quad &\mathbb{P}(\Omega(S)) - \mathbb{P}(\Omega(S \cup \{\bar{t}\})) = 0 \\ \Leftrightarrow \quad &\varphi(S) - \varphi(S \cup \{\bar{t}\}) = 0 \end{aligned}$$

Because the scenario $\bar{t} \in T$ was chosen arbitrarily the candidate-condition is fulfilled.
All together the claim holds. □

After defining a suitable stopping criterion for an iterative discretization scheme, we can ask how the probust terms approximation behaves for $k \to \infty$. The answer in the case of increasing discretizations $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$ is:

Theorem 2.1.11 (Convergence of increasing discretization sequences)
Let the inner function g be continuous w.r.t. $(\xi, t) \in \Xi \times T$, Assumption 2.1.6 be satisfied and let be $(T_k)_{k\in\mathbb{N}}, T_k \subseteq T$ for all $k \in \mathbb{N}$ be a sequence of subsets of T such that $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$.
Then we can conclude

$$\lim_{k\to\infty} \varphi(T_k) = \varphi(\lim_{k\to\infty} T_k).$$

If additionally $\lim_{k\to\infty} T_k$ fulfills the candidate-condition, then we know

$$\lim_{k\to\infty} \varphi(T_k) = \varphi(T).$$

Proof. We show the claim directly by the definition of φ and the dominated convergence theorem:
Be aware that $\varphi(S) = \int_\Xi \chi_{\Omega(S)}(\xi) d\mathbb{P}(\xi)$ holds for any subset $S \subseteq T$, where χ_A is the indicator function of a set $A \in \mathcal{A}$. Let us denote $S := \lim_{k\to\infty} T_k = \bigcup_{k\in\mathbb{N}} T_k$. We have to show point-wise convergence of $\chi_{\Omega(T_k)}$ to $\chi_{\Omega(S)}$ for $k \to \infty$ on Ξ to use the dominated convergence theorem. Please mind that χ is bounded by 1 by definition.
For this, we fix an arbitrary $\bar{\xi} \in \Xi$ and use a case-distinction:
First case $\bar{\xi} \in \Omega(S)$: If we assume $\bar{\xi} \in \Omega(S)$, we know $\chi_{\Omega(S)}(\bar{\xi}) = 1$ and by $T_k \subseteq S$ that $\Omega(S) \subseteq \Omega(T_k)$ for all $k \in \mathbb{N}$. Consequently, $\chi_{\Omega(S)}(\bar{\xi}) = 1$ implies $\chi_{\Omega(T_k)}(\bar{\xi}) = 1$ for all $k \in \mathbb{N}$ which ensures point-wise convergence.
Second case $\bar{\xi} \notin \Omega(S)$: If we assume $\bar{\xi} \notin \Omega(S)$, then we know $\chi_{\Omega(S)}(\bar{\xi}) = 0$ and there exists some $\bar{s} \in S$ such that $g(\bar{\xi}, \bar{s}) > 0$. By definition of S we know that there exists a $N \in \mathbb{N}$ such that $\bar{s} \in T_N$. By $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$, we know that $s \in T_k$ for all $k \geq N$. Consequently, $\max_{t\in T_k} g(\bar{\xi}, t) \geq g(\bar{\xi}, \bar{s}) > 0$ and $\bar{\xi} \notin \Omega(T_k)$ for all $k \geq N$. This implies $\lim_{k\to\infty} \chi_{\Omega(T_k)}(\bar{\xi}) = 0$.
Since χ_A does not have any other values than 0 and 1 the point-wise convergence holds. As all sets $\Omega(S), \Omega(T_k)$ with $k \in \mathbb{N}$ are measurable due to Assumption 2.1.1, we can use the dominated convergence theorem to conclude

$$\lim_{k\to\infty} \int_\Xi |\chi_{\Omega(T_k)}(\xi) - \chi_{\Omega(S)}(\xi)| d\mathbb{P}(\xi) = 0$$

which implies

$$\lim_{k\to\infty} \varphi(T_k) = \lim_{k\to\infty} \int_\Xi \chi_{\Omega(T_k)}(\xi) d\mathbb{P}(\xi) = \lim_{k\to\infty} \int_\Xi \chi_{\Omega(S)}(\xi) d\mathbb{P}(\xi) = \varphi(S).$$

This shows the first part of the claim.
If S satisfies the candidate-condition, we know by Theorem 2.1.10 that $\varphi(S) = \varphi(T)$. This shows the second part of the claim. □

We emphasize that the candidate-condition has not to be fulfilled to ensure convergence of the sequence $(\varphi(T_k))_{k\in\mathbb{N}}$, it however has to be fulfilled to guarantee convergence to $\varphi(T)$ as the choice $T_k = \emptyset$ for all $k \in \mathbb{N}$ implies.

In Section 2.2 we are interested in discretization strategies that do not necessarily increase in every iteration. Nevertheless, we want to guarantee convergence using these strategies. Consequently, we have to think about how to replace the limes of a monotonically increasing subset sequence.

In the next theorem we show that the limes inferior created by an arbitrary subset sequence $(T_k)_{k\in\mathbb{N}}, T_k \subseteq T$ for all $k \in \mathbb{N}$ is an appropriate choice.

Theorem 2.1.12 (Convergence of arbitrary discretization sequences)
Let the inner function g be continuous w.r.t. $(\xi, t) \in \Xi \times T$, Assumption 2.1.1 be satisfied and let $(T_k)_{k\in\mathbb{N}}$ with $T_k \subseteq T$ for all $k \in \mathbb{N}$ be a sequence of subsets of T.
Define

$$S := \liminf_{k\to\infty} T_k = \bigcup_{n=1}^{\infty} \bigcap_{k=n}^{\infty} T_k$$

that fulfills the candidate-condition

$$\forall t \in T, \epsilon > 0 : \varphi(S) - \varphi(S \cup \{t\}) < \epsilon.$$

Then we can conclude

$$\limsup_{k\to\infty} \varphi(T_k) = \lim_{k\to\infty} \varphi(T_k) = \varphi(T).$$

Proof. We show the claim directly by the definition of the limes inferior:

Because we assumed that the candidate-condition is fulfilled, we know by Theorem 2.1.10 that

$$\varphi(S) = \varphi(T).$$

Therefore, we concentrate on estimating $\varphi(T_k)$ for all $k \in \mathbb{N}$ by

$$\varphi(S) \geq \limsup_{k\to\infty} \varphi(T_k) \geq \varphi(T). \tag{2.3}$$

Please note that $\varphi(\tilde{T}) \in [0, 1]$ for any subset $\tilde{T} \subseteq T$. Consequently, the sequence $(\varphi(T_k))_{k\in\mathbb{N}}$ is bounded and we know that there exists at least on accumulation point such that the limes superior is well-defined.

The second inequality in (2.3) holds by the assumption $T_k \subseteq T$ for all $k \in \mathbb{N}$ and Proposition 2.1.2.

Next, we proof the first inequality in (2.3) by considering the following sequence of sets:

$$S_k := \bigcap_{j\geq k} T_j \quad \forall k \in \mathbb{N}$$

By definition of S_k we know that for any fixed $k \in \mathbb{N}$

$$S_k = T_k \cap \bigcap_{j \geq k+1} T_j = T_k \cap S_{k+1}.$$

This implies

$$\begin{aligned}&S_k \subseteq T_k, S_k \subseteq S_{k+1} \text{ and}\\ &\lim_{k\to\infty} S_k = S.\end{aligned}$$

Using Proposition 2.1.2 and Theorem 2.1.11, we can conclude:

$$\begin{aligned}&\varphi(S_k) \geq \varphi(T_k) \quad \forall k \in \mathbb{N}\\ &\varphi(S) = \lim_{k\to\infty} \varphi(S_k) \geq \limsup_{k\to\infty} \varphi(T_k).\end{aligned}$$

Hence, the first inequality in (2.3) also holds and we can write

$$\varphi(T) = \varphi(S) \geq \limsup_{k\to\infty} \varphi(T_k) \geq \liminf_{k\to\infty} \varphi(T_k) \geq \varphi(T).$$

Therefore the limes of $(\varphi(T_k))_{k\in\mathbb{N}}$ is well-defined and the claim holds. □

Take note that the limes inferior does not give any information about single subsets $T_k, k \in \mathbb{N}$. It might happen that one special subset $T_{k^*}, k^* \in \mathbb{N}$ satisfies the candidate-condition and therefore $\varphi(T_{k^*}) = \varphi(T)$, but $T_{k^*} \cap T_k = \emptyset$ for all $k \geq k^*+1$ and consequently the limes inferior will not contain T_{k^*}.

One open question remains:
What is the benefit w.r.t. φ when adding $t \in T$ to the current discretization $S \subseteq T$?
This question will be answered in Lemma 2.3.4. Right now we have to settle for the general estimation

$$\varphi(T) \leq \varphi(S \cup \{t\}) \leq \varphi(S).$$

Nevertheless, after understanding which condition a discretization $S \subseteq T$ has to fulfill to be able to replace the whole set T in probust terms, how adding a new discretization points $t \in T$ to S influences $\varphi(S)$ and how we can work with subset sequences $(T_k)_{k\in\mathbb{N}}, T_k \subseteq T$ for all $k \in \mathbb{N}$, we can concentrate on discretized probust functions.

After we understood how discretization works with probust terms, we now focus on how discretization works with probust functions.
First of all, we have to assume that given a decision $x \in X$ and some subset $S \subseteq T$, we can evaluate $\varphi(x, S)$. Therefore, we make the following assumption.

Assumption 2.1.13 (Well-posedness of $\varphi(x,S)$)
Given compact sets X, T, a probability space $(\Xi, \mathcal{A}, \mathbb{P})$, a function $g : X \times \Xi \times T \to \mathbb{R}$ and a probust function $\varphi : X \to [0,1], x \mapsto \varphi(x) = \mathbb{P}(g(x,\xi,t) \le 0 \ \forall t \in T)$, we assume that $\Xi(x,S) := \{\xi \in \Xi \mid g(x,\xi,t) \le 0 \ \forall t \in S\} \in \mathcal{A}$ for all $x \in X, S \subseteq T$.

With this assumption we can evaluate all terms $\varphi(x,S) := \mathbb{P}(g(x,\xi,t) \le 0 \ \forall t \in S)$ given $x \in X$ and $S \subseteq T$. Analogously to Proposition 2.1.8, we can show that $\varphi(\cdot,S)$ changes continuously w.r.t. $x \in X$ for all $S \subseteq T$.

Proposition 2.1.14 ($\varphi(\cdot,S)$ is continuous)
Given that Assumption 2.1.13 holds and that the inner function g is continuous w.r.t. $(x,\xi,t) \in X \times \Xi \times T$ satisfying $\mathbb{P}\left(\max_{s \in \mathrm{cl}(S)} g(x,\xi,s) = 0\right) = 0$ and $S \subseteq T$, then the function $\varphi(\cdot,S) : X \to [0,1]$ is continuous.

Proof. We show the claim directly by using Theorem 1.2.8 and Theorem 1.1.3:
If $S = \emptyset$, then $\varphi(x,S) = 1$ for all $x \in X$ is continuous w.r.t. $x \in X$ as a constant function. Otherwise, choose a nonempty subset $S \subseteq T$ arbitrarily. Then we can focus on the expression

$$\varphi(\cdot,S) \quad = \quad \mathbb{P}\left(\max_{s \in \mathrm{cl}(S)} g(\cdot,\xi,s) \le 0\right).$$

Because $\mathrm{cl}\,(S) \subseteq T$ is a closed subset of a compact set, it is compact itself. By Berge's maximum theorem the function

$$g(x,\xi,S) = \max_{s \in \mathrm{cl}(S)} g(x,\xi,s)$$

is continuous in $(x,\xi) \in X \times \Xi$ and by Theorem 1.1.3 we know that $\varphi(\cdot,S)$ is also continuous. As $S \subseteq T$ was chosen arbitrarily, the claim holds. □

By what we have seen so far in this section, we want to focus on subsets $S \subseteq T$ that fulfill the candidate-condition. As probust functions can be interpreted as probust terms that vary with $x \in X$, it might happen that a subset $S \subseteq T$ fulfills the candidate-condition at $x_1 \in X$ does not fulfill it at $x_2 \in X$ as the next example shows.

Example 2.1.15 (bottle-neck-condition over different decisions)
Consider the probust function defined by the probability space $(\mathbb{R}, \mathcal{B}, \mathbb{P})$, where $\mathcal{B}$ is the Borel-σ-algebra on $\mathbb{R}$ and $\mathbb{P}$ is the probability measure implied by $Z \sim \mathcal{U}([0,1])$. Furthermore, set $X := [-1,1], T := [-1,1]$ and $g(x,\xi,t) := -xt + \xi$ for all $(x,\xi,t) \in X \times \Xi \times T$. For $x_1 = 1$ we can find the (unique) maximizer of $g(x_1,\xi,t)$ within T that is $t_1^* = -1$ for all $\xi \in \Xi$. Consequently, $S = \{-1\}$ fulfills the bottle-neck-condition (and also the candidate-condition by Theorem 2.1.10) for $x_1 = 1$.
Considering $x_2 = -1$, we know that $t_2^* = 1$ is the unique maximizer of $g(x_2,\xi,t)$ within T for all $\xi \in \Xi$ and $S = \{-1\}$ does not fulfill the bottle-neck-condition (or equivalently the candidate-condition) for $x_2 = -1$ because for fixed $\overline{\xi} = 1$ the condition $g(x_2,\overline{\xi},-1) \le 0$ does not imply $g(x_2,\overline{\xi},1) \le 0$.

Therefore, we define a special subset of X that aggregates all decisions for which a given subset $S \subseteq T$ fulfills the candidate-condition.

Definition 2.1.16 (S-candidate-points)
Given a probust function $\varphi : X \to [0,1]$ with continuous inner function $g : X \times \Xi \times T \to \mathbb{R}$ and a subset $S \subseteq T$, we define the set of S-candidate-points as

$$C(S) := \{x \in X \mid S \textit{ fulfills the candidate-condition for } \varphi(x)\}.$$

The next proposition gives sufficient conditions for a set of S-candidate points to be compact. We need this property to talk about convergence of $(\varphi(\cdot, T_k))_{k\in\mathbb{N}}$ as a function afterwards.

Proposition 2.1.17 ($C(S)$ is compact)
Given Assumption 2.1.13 *with continuous inner function g, then the set of S-candidate-points is compact.*

Proof. We show the claim directly by Theorem 2.1.10 and Proposition 2.1.14:
Because g is continuous w.r.t. $(\xi, t) \in \Xi \times T$ for all $x \in X$ by assumption, we can use Theorem 2.1.10 to rewrite the set of S-candidate points as

$$C(S) = \{x \in X \mid \varphi(x, S) - \varphi(x, T) = 0\}.$$

As S and T are fixed sets, we know by the continuity of g w.r.t. $(x, \xi, t) \in X \times \Xi \times T$ and by Proposition 2.1.14 that $\varphi(\cdot, S) - \varphi(\cdot, T)$ is a continuous function w.r.t. $x \in X$. Consequently, its pre-image of $\{0\}$ is closed. As $X \subseteq \mathbb{R}^n$ was assumed to be compact, we know that $C(S)$ is compact as an intersection of a closed set with a compact set. Therefore, the claim holds. □

Take into account that $C(S) \subseteq X$ is compact, but it might be empty.
To guarantee that a given subset sequence $(T_k)_{k\in\mathbb{N}}$ with $T_k \subseteq T$ for all $k \in \mathbb{N}$ has a non-empty set of S-candidate points $C(S)$, where $S := \liminf_{k\to\infty} T_k$, will be the main part of constructing discretization schemes in Section 2.3.
In the next proposition, we see how the convergence of $T_k \to S$ influences the convergence of $\varphi(\cdot, T_k)$ as a function of $x \in X$ if we guarantee increasing subsets $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$:

Proposition 2.1.18 (Convergence of $\varphi(\cdot, T_k)$ towards $\varphi(\cdot, S)$)
Given that Assumption 2.1.13, the setting of Theorem 2.1.11 and a sequence of subsets $(T_k)_{k\in\mathbb{N}}, T_k \subseteq T_{k+1} \subseteq T$ for all $k \in \mathbb{N}$, we can conclude that $\varphi(\cdot, T_k) \to \varphi(\cdot, S)$ uniformly w.r.t. $x \in X$, that means

$$\lim_{k\to\infty} \max_{x\in X} |\varphi(x, T_k) - \varphi(x, S)| = 0,$$

where $S := \lim_{k\to\infty} T_k = \bigcup_{k=1}^{\infty} T_k$.

Proof. We prove this claim directly using Dini's theorem.
We know by Proposition 2.1.14 that $\varphi(\cdot, T_k)$ for all $k \in \mathbb{N}$ and $\varphi(\cdot, S)$ are continuous functions.
To show point-wise convergence of $(\varphi(x, T_k))_{k\in\mathbb{N}}$ for $x \in X$, we fix an arbitrary $\overline{x} \in X$. Because we assumed $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$, $(\varphi(\overline{x}, T_k))_{k\in\mathbb{N}}$ is a monotonically decreasing sequence in $[0, 1]$. Therefore, we know that $\varphi(\overline{x}, T_k)$ converges to some value $\varphi_{\overline{x}} \in [0, 1]$. Because of Theorem 2.1.10 and Assumption 2.1.13 we know that

$$\varphi(\overline{x}, S) = \varphi(\overline{x}, \lim_{k\to\infty} T_k) = \lim_{k\to\infty} \varphi(\overline{x}, T_k) = \varphi_{\overline{x}}.$$

As $\overline{x} \in X$ was chosen arbitrarily, $(\varphi(\cdot, T_k))_{k\in\mathbb{N}}$ converges point-wise to $\varphi(\cdot, S)$. Eventually, we can use Dini's theorem to conclude uniform convergence of $(\varphi(\cdot, T_k))_{k\in\mathbb{N}}$ to $\varphi(\cdot, S)$ and the claim is fulfilled. □

After showing how $\varphi(\cdot, T_k)$ converges for $k \to \infty$ towards $\varphi(\cdot, S)$, the question arises for which $x \in X$ we can conclude $\lim_{k\to\infty} \varphi(x, T_k) = \varphi(x, T)$. The next lemma provides the answer:

Lemma 2.1.19 (Convergence of $\boldsymbol{\varphi(\cdot, T_k)}$ towards $\boldsymbol{\varphi(\cdot, T)}$)
Given Assumption 2.1.13 *and a sequence of subsets* $(T_k)_{k\in\mathbb{N}} \subseteq 2^T$. *We consider the set* $S := \liminf_{k\to\infty} T_k$ *and can conclude that* $(\varphi(\cdot, T_k))_{k\in\mathbb{N}}$ *converges uniformly to* $\varphi(\cdot, T)$ *on* $C(S)$*, meaning*

$$\lim_{k\to\infty} \max_{x\in C(S)} |\varphi(x, T_k) - \varphi(x, T)| = 0.$$

Proof. We show this claim directly by using Lemma 2.1.19 and Theorem 2.1.12:
By Lemma 2.1.19 we know that $(\varphi(\cdot, T_k))_{k\in\mathbb{N}}$ converges uniformly to $\varphi(\cdot, S)$ on X and therefore especially on $C(S) \subseteq X$. By definition of $C(S)$ and Theorem 2.1.12 we know that $\varphi(x, S) = \varphi(x, T)$ for all $x \in C(S)$. Therefore the claim holds. □

We have seen which setting is sufficient to guarantee the well-posedness of $\varphi(x, S)$ for $x \in X, S \subseteq T$ and how the function sequence $\varphi(\cdot, T_k)$ converges uniformly towards $\varphi(\cdot, S)$ on X. Furthermore, we characterized the points, where $\varphi(\cdot, T_k)$ converges uniformly to $\varphi(\cdot, T)$, namely at $x \in C(S)$, where $S = \liminf_{k\to\infty} T_k$.
In the next section, we define the probust discretization algorithm that generates a sequence $(x_k)_{k\in\mathbb{N}} \subseteq X$ to a given sequence of subsets $(T_k)_{k\in\mathbb{N}} \subseteq 2^T$ such that accumulation points of $(x_k)_{k\in\mathbb{N}}$ solve the (standard) probust optimization problem

$$\text{SPP}_S : \min_{x\in X} f(x) \text{ s.t. } \mathbb{P}(g(x, \xi, t) \leq 0 \ \forall t \in S) \geq p,$$

where again $S = \liminf_{k\to\infty} T_k$.

2.2 The probust discretization algorithm

With the knowledge of how to discretize probust functions, we are ready to define the probust discretization algorithm in this section and show its convergence given an appropriate subset sequence $(T_k)_{k\in\mathbb{N}} \subseteq 2^T$.

To formulate the probust discretization algorithm, we have to introduce some quantities beforehand:

Definition 2.2.1 (Subset schemes Φ)
We call the function sequence $\Phi = (\Phi_k)_{k\in\mathbb{N}}$ a subset scheme, if for all $k \in \mathbb{N}$ the functions $\Phi_k : X \times 2^T \to 2^T$ are well-defined.
If Φ is a subset scheme and for all $k \in \mathbb{N}$ the functions Φ_k satisfy $|\Phi_k(x, S)| < \infty$ for all $x \in X, S \in 2^T$ with $|S| < \infty$, we call Φ a discretization scheme.
If Φ is a subset scheme and for all $k \in \mathbb{N}$ the functions Φ_k does not depend on X or S, we call Φ a predefined subset scheme. Otherwise, it is called an adaptive subset scheme.
If Φ is a subset scheme satisfying $\Phi_k(x, S) \supseteq S$ for all $x \in X, S \subseteq T, k \in \mathbb{N}$, we call Φ an increasing subset scheme.

In the following algorithm, the subset scheme Φ defines (well-defined) rules on how to use the current iterates $x_k \in X$ and $T_k \subseteq T$ to define a new subset $T_{k+1} \subseteq T$ for the next iteration.

Algorithm 3 Probust discretization algorithm

```
1: Inputs:
       Standard probust optimization problem instance SPP, discretization
       scheme $\Phi$, initial decision $x_0 \in X$, initial discretization $T_0 \subseteq T$
2: Initialize:
       $k := 0$
3: do
4:     $x_{k+1} \leftarrow \arg\min_{x\in X} f(x)$ s.t. $\varphi(x, T_k) \geq p$
5:     $T_{k+1} \leftarrow \Phi_k(x_{k+1}, T_k)$
6:     $k \leftarrow k + 1$
7: while stopping criterion is not fulfilled
8: Results:
       Sequence $(x_k, T_k)_{k\in\mathbb{N}}$
```

Please note that the optimization problem in line 4 is a JCC if all subsets $T_k, k \in \mathbb{N}$ are finite. Then we can use Theorem 1.1.3 and Corollary 1.1.5 to ask if the solution is well-defined and unique in each iteration $k \in \mathbb{N}$.
The definition of a stopping criterion in line 7 will be handled in Chapter 4.

Before we study the convergence of the given algorithm, we need one additional lemma that guarantees decreasing feasible sets of probust optimization problems for increasing subsets $(T_k)_{k\in\mathbb{N}}$, where increasing and decreasing is meant w.r.t. the partial ordering $\subseteq$.

Lemma 2.2.2 (Decreasing feasible sets for $S \subseteq T$)
Given the setting of Theorem 1.3.9 *and a subset $S \subseteq T$, we can conclude*

$$\mathcal{F}_{T,p} \subseteq \mathcal{F}_{S,p},$$

where $\mathcal{F}_{S,p} := \{x \in X \mid \varphi(x, S) \geq p\}$ is the feasible set of the (standard) probust optimization problem with subset $S \subseteq T$.

Proof. We show $\mathcal{F}_{T,p} \subseteq \mathcal{F}_{S,p}$ element-wise. For this, let us fix an arbitrary $\overline{x} \in \mathcal{F}_{T,p}$. By definition this implies $\varphi(\overline{x}, T) \geq p$. Using Proposition 2.1.2, we know by $S \subseteq T$ that $\varphi(\overline{x}, T) \leq \varphi(\overline{x}, S)$. Consequently, $\varphi(\overline{x}, S) \geq p$ and $\overline{x} \in \mathcal{F}_{S,p}$. As $\overline{x} \in X$ was fixed arbitrarily, the claim holds. □

The next theorem states one of the main results of this thesis: The convergence of the probust subset algorithm.

Theorem 2.2.3 (Convergence theorem)
Given the setting of Theorem 1.3.9 *and a subset scheme Φ which creates a sequence of subsets $(T_k)_{k\in\mathbb{N}} \subseteq 2^T$ such that for any accumulation point $\bar{x}$ of $(x_k)_{k\in\mathbb{N}}$ the limes inferior of the corresponding subsequence of subsets $S = \liminf_{j\to\infty} T_{k_j}$ with $\lim_{j\to\infty} x_{k_j} = \bar{x}$ satisfies $\bar{x} \in C(S)$. Then $\bar{x}$ is a minimizer of the original probust optimization problem.*

Proof. We prove the statement by showing that the accumulation point $\bar{x}$ is feasible and optimal w.r.t. the original probust optimization problem:
As the setting of Theorem 1.3.9 is given, we know that the probust optimization problem has a well-defined solution. Let $\bar{x} \in X$ be an accumulation point of the sequence $(x_k)_{k\in\mathbb{N}}$. Because X is compact, we can guarantee the existence of at least one of these accumulation points.
Without loss of generality we assume that $x_k \to \bar{x}$ for $k \to \infty$. Otherwise, there exists a subsequence $(x_{k_i})_{i\in\mathbb{N}}$ such that $k_{i+1} > k_i$ for all $i \in \mathbb{N}$ and $x_{k_i} \to \bar{x}$ for $i \to \infty$ that we would consider instead.

We show that $\bar{x}$ is feasible w.r.t. the original probust optimization problem indirectly by using Lemma 2.2.2, Fatou's lemma and Lemma 2.1.19 :
Assume that $\bar{x} \notin \mathcal{F}_{T,p}$. Because the minimizer x_k is well-defined we get for any $k \in \mathbb{N}$ that $x_k \in \mathcal{F}_{T_k,p}$ and therefore $\varphi(x_k, T_k) \geq p$ for any $k \in \mathbb{N}$.
With Lemma 2.2.2 we get

$$\begin{aligned} &\mathcal{F}_{S_{k+i},p} \subseteq \mathcal{F}_{S_k,p} \ \forall i, k \in \mathbb{N}, \\ &\text{where } S_j := \bigcap_{k\geq j} T_k \quad \forall j \in \mathbb{N}. \end{aligned}$$

Consequently, S_j fulfills $S_j = T_j \cap S_{j+1}$ for all $j \in \mathbb{N}$ and we can conclude for any fixed $i, k \in \mathbb{N}$:

$$\begin{aligned} &x_{k+i} \in \mathcal{F}_{T_{k+i},p} \subseteq \mathcal{F}_{S_{k+i},p} \subseteq \mathcal{F}_{S_k,p} \\ \Rightarrow\quad &p \leq \varphi(x_{k+i}, T_{k+i}) \leq \varphi(x_{k+i}, S_{k+i}) \leq \varphi(x_{k+i}, S_k) \end{aligned}$$

Next, we rewrite the last inequality as:

$$\begin{aligned}
p &\leq \varphi(x_{k+i}, S_k) \\
&= \mathbb{P}_\xi(\Omega\,(x_{k+i}, S_k))) \\
&= \int_{\xi\in\Omega(x_{k+i},S_k)} 1 d\mathbb{P}(\xi) \\
&= \int_{\xi\in\Xi} \chi_{\Omega(x_{k+i},S_k)}(\xi) d\mathbb{P}(\xi) \\
&= \int_{\xi\in\Xi} \chi_{\max_{t\in\mathrm{cl}(S_k)} g(x_{k+i},\cdot,t)\leq 0}(\xi) d\mathbb{P}(\xi)
\end{aligned}$$

Using Fatou's lemma and the continuity of g w.r.t. x we conclude further:

$$\begin{aligned}
p &\leq \limsup_{i\to\infty} \int_{\xi\in\Xi} \chi_{\max_{t\in\mathrm{cl}(S_k)} g(x_{k+i},\cdot,t)\leq 0}(\xi) d\mathbb{P}(\xi) \\
&\leq \int_{\xi\in\Xi} \limsup_{i\to\infty} \chi_{\max_{t\in\mathrm{cl}(S_k)} g(x_{k+i},\cdot,t)\leq 0}(\xi) d\mathbb{P}(\xi) \\
&= \int_{\xi\in\Xi} \chi_{\max_{t\in\mathrm{cl}(S_k)} g(\bar{x},\cdot,t)\leq 0}(\xi) d\mathbb{P}(\xi) \\
&= \varphi(\bar{x}, S_k)
\end{aligned}$$

Therefore, $\bar{x} \in \mathcal{F}_{S_k,p}$ for every $k \in \mathbb{N}$ or equivalently $\bar{x} \in \bigcap_{k\in\mathbb{N}} \mathcal{F}_{S_k,p}$.
Because we assumed that $S = \liminf_{k\to\infty} T_k = \lim_{k\to\infty} S_k$ fulfills $\bar{x} \in C(S)$, we know by Lemma 2.1.19 and the definition of $C(S)$ that we can write

$$p \leq \lim_{k\to\infty} \varphi(\bar{x}, S_k) = \varphi(\bar{x}, S) = \varphi(\bar{x}, T).$$

Consequently, $\bar{x}$ is feasible w.r.t. the original problem.

Next we show that $\bar{x}$ is a minimizer of the original probust optimization problem indirectly:
Let us assume that there exists $x^* \in \mathcal{F}_{T,p}$ with $f(x^*) < f(\bar{x})$. Because $\mathcal{F}_{T,p} \subseteq \mathcal{F}_{T_k,p}$ for all $k \in \mathbb{N}$, we know that $x^*, \bar{x} \in \mathcal{F}_{T_k,p}$ for all $k \in \mathbb{N}$. Because of the definition of x_k in the probust discretization algorithm 3 line 4 and the continuity of f w.r.t. $x \in X$ we get the following contradiction:

$$\begin{aligned}
f(x^*) &< f(\bar{x}) = \lim_{k\to\infty} f(x_k) \\
&= \lim_{k\to\infty} \min_{x\in\mathcal{F}_{T_k,p}} f(x) \\
&\leq \lim_{k\to\infty} \min_{x\in\mathcal{F}_{T,p}} f(x) \\
&= \min_{x\in\mathcal{F}_{T,p}} f(x) \leq f(x^*)
\end{aligned}$$

Therefore, our assumption is wrong what proves the optimality of $\bar{x}$ within $\mathcal{F}_{T,p}$.
All together the accumulation point $\bar{x}$ is feasible and minimal and therefore a minimizer of the original probust optimization problem. □

In the last proof, we have seen that the difficult part is showing that the candidate-condition holds at an accumulation point $\bar{x}$.
The part of the proof focusing on the optimality of the iterates $\bar{x}$ is rather short. This can be used to soften the definition of the iterates $x_k, k \in \mathbb{N}$ to be nearly optimal solutions of the corresponding subproblems as shown by Theorem 1 in the paper of Berthold, Heitsch, Henrion and Schwientek [15].
If we use the last theorem on an increasing subset sequence $(T_k)_{k\in\mathbb{N}}$ with $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$, we can consider $S = \bigcup_{k=1}^{\infty} T_k$. Then, we just have to check, if S satisfies the candidate-condition for the accumulation point $\bar{x} \in X$. The next example shows that checking the candidate-condition for an arbitrary sequence of subsets might lead to unsatisfying results:

Example 2.2.4 ($\liminf_{k\to\infty} T_k$ is too weak in general)
Consider the probust optimization problem

$$\min_{x\in[0,2]} -x \text{ s.t. } \mathbb{P}(-x+\xi+t^2 \le 0 \text{ for all } t \in T := [-1,1]) \ge 0.9, Z \sim \mathcal{U}([0,1]) .$$

Then using the discretization scheme Φ defined by

$$\Phi_k : X \times 2^T \to 2^T, (x,S) \mapsto \begin{cases} \{-1\} & \text{if } k \text{ is even and} \\ \{1\} & \text{if } k \text{ is odd} \end{cases}$$

leads to the iterates $x_k = x^* = 1.9$.
Consequently, the generated iterates converge towards the minimizer of the original probust optimization problem. Nevertheless, the set $S = \liminf_{k\to\infty} T_k$ is empty and does not fulfill $\varphi(x_k, S) = \varphi(x_k, T)$ for any $k \in \mathbb{N}$.
On the contrary, every $T_k, k \in \mathbb{N}$ fulfills $\varphi(x_k, T_k) = \varphi(x_k, T)$. Therefore, we must not only focus on the limes inferior of the whole sequence $(T_k)_{k\in\mathbb{N}}$, but also consider the limes inferior of subsequences of $(T_k)_{k\in\mathbb{N}}$. Note that the subsequence $(x_{2k}, T_{2k})_{k\in\mathbb{N}}$ fulfills the condition $\lim_{k\to\infty} \varphi(x_{2k}, T_{2k}) = \varphi(x^*, T)$.

The last theorem can be used as a tool to show directly that a given subset scheme Φ leads to a solution of the original probust optimization problem. It seems especially useful if we work with increasing discretization schemes. If we do not ensure increasing subset schemes, the proof of convergence might be more difficult as we have to specify subsequences (of subsets) of subsequences (of iterates) to guarantee convergence as the last example pointed out. This motivates us to focus on another approach that can be used by comparing two subset schemes Φ_1 and Φ_2.
We want to show that if the scheme Φ_1 leads to iterates that converge to a minimizer of the original probust optimization problem and Φ_2 generates "better" subsets than Φ_1, then the iterates generated by the probust subset algorithm using Φ_2 do also converge to a minimizer of the original probust optimization problem.
With this approach we switch the perspective from the iterates $x_k, k \in \mathbb{N}$ to the minimizer $x^* \in X$. We interpret the subsets $(T_k)_{k\in\mathbb{N}}$ generated by Φ_1 and $(S_k)_{k\in\mathbb{N}}$ generated by Φ_2 as measures how precise Φ_1 and Φ_2 are able to describe the feasibility of x^* by approximating $\varphi(x^*, T)$.

Lemma 2.2.5 (Reduction of subset schemes 1)
Assume that we have a subset scheme Φ_1 that generates a sequence $(T_k)_{k\in\mathbb{N}}$ and denote $S_1 := \liminf_{k\to\infty} T_k$. Assume further that a the discretization algorithm with subset scheme Φ_2 creates a sequence $(x_k, S_k)_{k\in\mathbb{N}}$ such that $(x_k)_{k\in\mathbb{N}}$ has a accumulation point $\bar{x} \in C(S_1)$ and the corresponding limes inferior $S_2 := \liminf_{j\to\infty} S_{k_j}$ satisfies $\bar{x} \in C(S_1, S_2) := \{x \in X \mid \varphi(x, S_1) = \varphi(x, S_2)\}$. Then $\bar{x}$ is an optimal solution of the original probust optimization problem.

Proof. We argue directly that any accumulation point $\bar{x}$ of $(x_k)_{k\in\mathbb{N}}$ fulfills $\bar{x} \in C(S_2)$ and therefore we can use Theorem 2.2.3 to conclude the claim.

Let us fix an arbitrary accumulation point $\bar{x}$ of $(x_k)_{k\in\mathbb{N}}$. As $\bar{x} \in C(S_1) \cap C(S_1, S_2)$ by assumption, we know by definition that

$$\varphi(\bar{x}, S_2) = \varphi(\bar{x}, S_1) = \varphi(\bar{x}, T).$$

With Theorem 2.1.10 we know that this implies $\bar{x} \in C(S_2)$.
Consequently Theorem 2.2.3 guarantees us that $\bar{x}$ is a solution of the original probust optimization problem. □

Be aware that this reduction lemma has two difficulties:
First, we have to show $\bar{x} \in C(S_1, S_2)$. As $S_2 \subseteq T$ might be given just implicitly, the set $C(S_1, S_2)$ is hard to compute. Consequently, this approach is useful if we know a lot about the subsets defined by Φ_1, e.g., if Φ_1 is an uniform discretization scheme.
Second, we have to guarantee that $\bar{x} = \lim_{j\to\infty} x_{k_j} \in C(S_1)$. Again, it would be nice to have a reference subset scheme, where we know what $C(S_1)$ looks like. We will see in the next section that the uniform discretization scheme is such a nice reference scheme as it guarantees $C(S_1) = X$.
The next theorem shows that we can exchange the condition $\bar{x} \in C(S_1, S_2)$ by a condition that works on the subset sequences $(T_k)_{k\in\mathbb{N}}$ generated by Φ_1 and $(x_k, S_k)_{k\in\mathbb{N}}$ generated by Φ_2.

Theorem 2.2.6 (Reduction of subset schemes 2)
Assume that we have a subset scheme Φ_1 that generates a sequence $(T_k)_{k\in\mathbb{N}}$ and denote $S := \liminf_{k\to\infty} T_k$. Assume further that a the discretization algorithm with subset scheme Φ_2 creates a sequence $(x_k, S_k)_{k\in\mathbb{N}}$ such that $(x_k)_{k\in\mathbb{N}}$ has an accumulation point $\bar{x} \in C(S)$ and for the subsequence $(x_{k_j})_{j\in\mathbb{N}}$ with $\lim_{j\to\infty} x_{k_j} = \bar{x}$ there exists some $N \in \mathbb{N}$ such that for all $j \in \mathbb{N}, j \geq N$ there exists $n_j, m_j \in \mathbb{N}, n_j, m_j \geq j$ such that $\varphi(x_{k_{n_j}}, S_{k_{m_j}}) \leq \varphi(x_{k_{n_j}}, T_j)$, then $\bar{x}$ is a minimizer of the original probust optimization problem.

Proof. We provide this proof directly using Propositions 2.1.14, 2.1.18 and Theorem 2.2.3. By our assumptions and Proposition 2.1.2, 2.1.14 and 2.1.18 we can estimate:

$$\begin{aligned}\varphi(\bar{x}, T) &= \lim_{j\to\infty} \varphi(x_{k_{n_j}}, T)\\ &\leq \lim_{j\to\infty} \varphi(x_{k_{n_j}}, S_{k_{m_j}})\\ &\leq \lim_{j\to\infty} \varphi(x_{k_{n_j}}, T_j)\\ &\leq \varphi(\bar{x}, \liminf_{j\to\infty} T_j)\\ &= \varphi(\bar{x}, T)\end{aligned}$$

Consequently, we know that $\bar{x} \in C(\liminf_{j\to\infty} S_{k_{m_j}})$ where $\lim_{j\to\infty} x_{k_{m_j}} = \bar{x}$. Using Theorem 2.2.3 the claim holds. □

Please mind that the iterate sequence generated by the probust discretization algorithm using Φ_1 does not appear anywhere in the last theorem.
In the next section, we specify some discretization strategies Φ and show that the corresponding subset sequences $(T_k)_{k\in\mathbb{N}}$ fulfill the candidate-condition for all generated accumulation points of $(x_k)_{k\in\mathbb{N}}$.

2.3 Examples of converging discretization schemes

The last section provides results to guarantee the convergence of the iterates generated by the probust subset algorithm 3 towards the minimizer of the corresponding (standard) probust optimization problem. Now, we use these results to show that discretization methods from semi-infinite optimization can also be modified to work in the probust optimization context. We focus on the uniform discretization and an adaptive discretization scheme from Blankenship and Falk [17].

First, we utilize Theorem 2.2.3 to show that the following subset schemes lead to a minimizer of the original probust optimization problem:

1. An uniform discretization scheme

2. An adaptive discretization scheme using inner function evaluation information called low-level adaptive discretization (LLAD) approach

3. An adaptive discretization scheme using probust function evaluation information called high-level adaptive discretization (HLAD) approach.

These discretization schemes define the basic schemes that we use in Part II of this thesis to create more specified subset schemes to solve the corresponding applications.

After considering the basic schemes, we finish this chapter by considering a non-increasing subset scheme that we compare with an uniform discretization scheme and use the reduction theorem 2.2.6 to ensure its convergence.

Uniform discretization scheme

The first discretization scheme $\Phi = (\Phi_k)_{k\in\mathbb{N}}$ which we analyze is an uniform discretization scheme with vanishing grid size defined by

$$\Phi_k : X \times 2^T \to 2^T, (x, S) \mapsto \left\{ t \in T \subseteq \mathbb{R}^q \mid \exists j \in \mathbb{Z}^q : t = \sum_{i=1}^{q} d_k j_i e_i \right\}, \tag{2.4}$$

where $e_i, i \in \{1, ..., q\}$ with $q \in \mathbb{N}$ is the i.th column of the identity matrix $I \in \mathbb{R}^{q\times q}$ and $(d_k)_{k\in\mathbb{N}}$ is a sequence of positive numbers converging to zero. By definition $|\Phi_k(x, S)| < \infty$ and $\Phi_k(x, S) \subsetneq \Phi_{k+1}(x, S)$ for all $k \in \mathbb{N}, x \in X, S \subseteq T$ and compact $T \subseteq \mathbb{R}^q$. Consequently, this scheme is a predefined discretization scheme. If the grid sizes $d_k, k \in \mathbb{N}$ are multiples of each other, the corresponding subset scheme is also an increasing subset scheme.

Lemma 2.3.1 (Convergence of uniform discretization scheme)
Let T be a set without isolated points, meaning that for all $t \in T$ and $r > 0$ there exists some $s \in T, s \neq t$ such that $s \in B_r(t)$. Additionally, let Φ be the uniform discretization scheme, then any accumulation point of the sequence $(x_k)_{k\in\mathbb{N}}$ generated by the probust subset algorithm 3 converges to a minimizer of the original probust optimization problem. Furthermore, $S := \lim_{k\to\infty} T_k$ satisfies $C(S) = X$.

Proof. We prove this claim by using Theorem 2.2.3:
By definition of Φ_k the set $T_{k+1} := \Phi_k(x_k, T_k)$ is well-defined for all $k \in \mathbb{N}$. Consequently, the algorithm creates iterates for all $k \in \mathbb{N}$ and as $X \subseteq \mathbb{R}^n$ was assumed to be compact, we know that $(x_k)_{k\in\mathbb{N}}$ has at least one accumulation point.
Let us fix an arbitrary accumulation point $\bar{x}$ of $(x_k)_{k\in\mathbb{N}}$. To use Theorem 2.2.3, we have to show $\bar{x} \in C(S)$. For this, we show that the candidate-condition is fulfilled for any $x \in X$ including $\bar{x}$.

We argue indirectly and assume there exists $x \in X$ such that the candidate-condition does not hold

$$\exists t_f \in T, \epsilon > 0 : \varphi(x, S) - \varphi(x, S \cup \{t_f\}) \geq \epsilon.$$

Because of Theorem 2.1.10 and the definition of the bottle-neck-condition, we know that there exists $\Omega \in \mathcal{A}$ such that

$$\Omega = \{\xi \in \Xi \mid g(x, \xi, s) \leq 0 \text{ for all } s \in S \wedge g(x, \xi, t_f) > 0\} \text{ and}$$
$$\mathbb{P}(\Omega) \geq \epsilon.$$

Additionally, we know by Assumption 2.1.6 that there exists a set $\tilde{\Omega} \in \mathcal{A}$ such that

$$\begin{aligned}&\tilde{\Omega} = \{\xi \in \Xi \mid g(x,\xi,s) < 0 \text{ for all } s \in S \wedge g(x,\xi,t_f) > 0\} \text{ and}\\ &\mathbb{P}(\tilde{\Omega}) = \mathbb{P}(\Omega) \geq \epsilon.\end{aligned}$$

Because g is continuous w.r.t. $t \in T$ and $\tilde{\Omega} \neq \emptyset$ due to $\epsilon > 0$, there exists a realization $\xi_f \in \tilde{\Omega}$ and a radius $r > 0$ such that

$$\begin{aligned}g(x,\xi_f,t) &> 0 \quad \forall t \in B_r(t_f),\\ \text{while } \max_{s \in \mathrm{cl}(S)} g(x,\xi_f,s) &< 0.\end{aligned}$$

By the definition of T_k, we know that $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$ and therefore $S = \bigcup_{k=1}^{\infty} T_k$. As we assumed that the distance between discretization points d_k goes to 0 for $k \to \infty$, we can find some $N \in \mathbb{N}$ such that for all $k \geq N$ we can guarantee that $d_k < r$ and consequently we have at least one discretization point $\bar{t} \in T_k \cap B_r(t_f) \subseteq S \cap B_r(t_f)$. Because $\bar{t} \in S \cap B_r(t_f)$, this point has to fulfill

$$\begin{aligned}&g(x,\xi_f,\bar{t}) > 0,\\ \text{but } &g(x,\xi_f,\bar{t}) \leq \max_{s \in \mathrm{cl}(S)} g(x,\xi_f,s) < 0.\end{aligned}$$

This is a contradiction and thus, the candidate-condition holds for any $x \in X$, especially for any accumulation point $\bar{x}$ of $(x_k)_{k \in \mathbb{N}}$ and the claim holds due to Theorem 2.2.3. □

The assumption of T having no isolated point is important. If we do not ensure this property of T, the uniform discretization scheme does not have to find the minimizer of the probust optimization problem.
One might think of an example, where $T = \{\sqrt{2}\}$ and the uniform discretization scheme is defined by its grid sizes $d_k = \frac{1}{k}$ for all $k \in \mathbb{N}$. Since $T_k = T \cap G_{d_k} = \emptyset$ in each iteration $k \in \mathbb{N}$ this would imply $\varphi(x, T_k) = 1$ for all $x \in X, k \in \mathbb{N}$.

Low-level adaptive discretization approach (LLAD)

In this section, we specify the subset scheme $\Phi = (\Phi_k)_{k \in \mathbb{N}}$ in the probust discretization algorithm introduced in Section 2.2 as

$$\Phi_k : X \times 2^T \to 2^T, (x,S) \mapsto S \cup S_k^* \tag{2.5}$$

where $S_k^* := \bigcup_{\xi \in \dot{\Xi}_k} \{t^*(\xi)\}$, $t^*(\xi) \in \arg\max_{t \in T} g(x,\xi,t)$ is a set of worst-case scenarios, depending on the realizations $\xi \in \dot{\Xi}_k$, where $\emptyset \neq \dot{\Xi}_k \subseteq \Xi$ is any subset of realization from Ξ for each $k \in \mathbb{N}$. By definition this scheme is an increasing subset scheme. If $|\dot{\Xi}_k| < \infty$ for all $k \in \mathbb{N}$, this scheme is a discretization scheme.

With this scheme we select the scenarios $t_k^*(\xi)$ that are worst w.r.t. certain realizations $\xi \in \dot{\Xi}_k$ considering decision x_k in each iteration $k \in \mathbb{N}$.
An interesting questions is which discretization $\dot{\Xi}_k \subseteq \Xi$ to select in each iteration $k \in \mathbb{N}$. We show the convergence of the scheme for randomly picked realizations and non-empty subsets $\dot{\Xi}_k \neq \emptyset$ for all iterations $k \in \mathbb{N}$.

Lemma 2.3.2 (Convergence of LLAD)
Choosing Φ as the low level adaptive discretization scheme, where $\dot{\Xi}_k$ are non-empty and the realizations $\xi_k \in \dot{\Xi}_k$ are randomly i.i.d. chosen (according to random variable in the probust problem) for all $k \in \mathbb{N}$. Then, any accumulation point of $(x_k)_{k\in\mathbb{N}}$ of the corresponding probust discretization method is $\mathbb{P}$-almost surely a minimizer of the original probust optimization problem.
Furthermore, we can conclude with $S := \lim_{k\to\infty} T_k$ that

$$\mathbb{P}(x \in C(S)) = 1.$$

Proof. We prove this claim by using Theorem 2.2.3:
We show that the candidate-condition is fulfilled $\mathbb{P}$-almost surely for any point $x \in X$ and therefore especially for any accumulation point $\bar{x} \in X$ of $(x_k)_{k\in\mathbb{N}}$.
Be aware that the maximizer of $\arg\max_{t\in T} g(x_k, \xi, t)$ is well-defined for all $\xi \in \Xi, k \in \mathbb{N}$ because we made the assumption that T is compact and g is continuous w.r.t. $t \in T$ for all $(x, \xi) \in X \times \Xi$. Furthermore, because $X \subseteq \mathbb{R}^n$ with $n \in \mathbb{N}$ is compact, the sequence $(x_k)_{k\in\mathbb{N}}$ has at least one accumulation point.

Now we fix an arbitrary $x \in X$ and argue indirectly by assuming:

$$\exists t_f \in T, \epsilon > 0 : \varphi(x, S) - \varphi(x, S \cup \{t_f\}) \geq \epsilon$$

Because of Theorem 2.1.10 and the definition of the bottle-neck-condition, we know that there exists $\Omega \in \mathcal{A}$ such that

$$\begin{aligned}&\Omega = \{\xi \in \Xi \mid g(x, \xi, s) \leq 0 \text{ for all } s \in S \wedge g(x, \xi, t_f) > 0\},\\ &\mathbb{P}(\Omega) \geq \epsilon.\end{aligned}$$

Additionally, we know by Assumption 2.1.6 that there exists a set $\tilde{\Omega} \in \mathcal{A}$ such that

$$\begin{aligned}&\tilde{\Omega} = \{\xi \in \Xi \mid g(x, \xi, s) < 0 \text{ for all } s \in S \wedge g(x, \xi, t_f) > 0\},\\ &\mathbb{P}(\tilde{\Omega}) = \mathbb{P}(\Omega) \geq \epsilon.\end{aligned}$$

Since g is continuous w.r.t. $\xi \in \Xi$ and $\tilde{\Omega} \neq \emptyset$ due to $\epsilon > 0$, we can find a realization $\xi_f \in \tilde{\Omega}$ and a radius $r > 0$, such that

$$\begin{gathered} g(x, \xi, t_f) > 0 \text{ for all } \xi \in B_r(\xi_f),\\ \text{while } \max_{s\in \mathrm{cl}(S)} g(x, \xi, s) < 0 \text{ for all } \xi \in B_r(\xi_f).\end{gathered}$$

Due to $\xi \in \operatorname{supp}(\mathbb{P})$, we can conclude that

$$\varepsilon := \mathbb{P}(B_r(\xi_f)) > 0.$$

As we choose at least one realization in each iteration $k \in \mathbb{N}$ and the choice of new scenarios $\xi_k \in \dot{\Xi}_k$ is i.i.d., the probability of selecting a realization $\xi_k \in B_r(\xi_f)$ for any $k \in \mathbb{N}$ is:

$$\begin{aligned} & \mathbb{P}(\exists k \in \mathbb{N} : \xi_k \in B_r(\xi_f)) \\ = \; & 1 - \mathbb{P}(\forall k \in \mathbb{N} : \xi_k \notin B_r(\xi_f)) \\ = \; & \lim_{k \to \infty} 1 - (1-\varepsilon)^k \\ = \; & 1 \end{aligned}$$

Therefore, we $\mathbb{P}$-almost surely pick a $\omega \in B_r(\xi_f)$ at some iteration $k \in \mathbb{N}$. Assuming that we pick $\omega \in B_r(\xi_f)$ in iteration $K \in \mathbb{N}$, we know that we add

$$t_K^*(\omega) := \arg\max_{t \in T} g(x, \omega, t) \in T_{K+1} \subseteq \bigcup_{k=1}^{\infty} T_k = S$$

to our discretization. Because of

$$g(x, \omega, t_K^*(\omega)) = \max_{t \in T} g(x, \omega, t) \geq g(x, \omega, t_f),$$

we know that the uncertainty pair $(\omega, t_f) \in B_r(\xi_f) \times T$ has to fulfill

$$\begin{gathered} g(x, \omega, t_f) > 0 \text{ as } \omega \in B_r(\xi_f), \\ \text{while } g(x, \omega, t_f) \leq g(x, \omega, t_K^*(\omega)) \leq \max_{s \in \operatorname{cl}(S)} g(x, \omega, t) < 0. \end{gathered}$$

This is a contradiction. Since we have chosen the realization ξ_k randomly, our basic assumption is $\mathbb{P}$-almost surely wrong. Consequently, the candidate-condition holds for any decision $\mathbb{P}$-almost surely, especially for all accumulation points $\bar{x} \in X$ of $(x_k)_{k \in \mathbb{N}}$. Using Theorem 2.2.3 the claim is true. □

High-level adaptive discretization approach (HLAD)

In this section, we specify the subset scheme $\Phi = (\Phi_k)_{k \in \mathbb{N}}$ in the probust discretization algorithm introduced in Section 2.2 as

$$\Phi_k : X \times 2^T \to 2^T, (x, S) \mapsto S \cup S_k^* \tag{2.6}$$

where $S_k^* := \{t_1^*, ..., t_{n_k}^*\}$, $(t_1^*, ..., t_{n_k}^*) \in \arg\min_{(t_1, ..., t_{n_k}) \in T^{n_k}} \varphi(x, S \cup \{t_1, ..., t_{n_k}\})$ are the worst-case scenarios considering the probability of $x \in X$ given the current subset $S \subseteq T$ for each $k \in \mathbb{N}$ and $(n_k)_{k \in \mathbb{N}}$ is a sequence of natural numbers. By definition this scheme is an increasing subset scheme. If $|T_0| < \infty$ for all $k \in \mathbb{N}$, this scheme is a discretization scheme.

Lemma 2.3.3 (Convergence of HLAD)
Choosing Φ as the high level adaptive discretization scheme any accumulation point of $(x_k)_{k\in\mathbb{N}}$ of the corresponding probust subset method converges to a minimizer of the original probust optimization problem.

Proof. We prove this claim by using Theorem 2.2.3:
We show that the candidate-condition is fulfilled for any accumulation point $\bar{x} \in X$ of $(x_k)_{k\in\mathbb{N}}$ indirectly.
Please mind that with Assumption 2.1.6 and by the compactness of T the selected scenarios are well-defined for all $k \in \mathbb{N}$. Furthermore, due to the compactness of $X \subseteq \mathbb{R}^n$, the sequence $(x_k)_{k\in\mathbb{N}}$ has at least one accumulation point. Without loss of generality we assume that the whole sequence has just one accumulation point. Otherwise, we switch notations to the subsequence.

Now we fix the accumulation point $\bar{x} \in X$, set $S := \lim_{k\to\infty} T_k$ and assume that the candidate-condition does not hold:

$$\exists t_f \in T, \epsilon > 0 : \varphi(\bar{x}, S) - \varphi(\bar{x}, S \cup \{t_f\}) \geq \epsilon$$

We will rewrite this inequality using Propositions 2.1.18 and 2.1.14:

$$\begin{aligned}
0 < \epsilon &\leq \lim_{k\to\infty} \varphi(\bar{x}, T_k) - \varphi(\bar{x}, T_k \cup \{t_f\}) \\
&= \lim_{k\to\infty} \varphi(\bar{x}, T_k) - \varphi(x_k, T_k \cup \{t_f\}) + \underbrace{\varphi(x_k, T_k \cup \{t_f\}) - \varphi(\bar{x}, T_k \cup \{t_f\})}_{\to 0 \text{ by Proposition 2.1.18}} \\
&= \lim_{k\to\infty} \varphi(\bar{x}, T_k) - \varphi(x_k, T_k \cup \{t_f\}) \\
&= \lim_{k\to\infty} \varphi(\bar{x}, T_k) - \varphi(x_k, T_{k+1}) + \underbrace{\varphi(x_k, T_{k+1}) - \varphi(x_k, T_k \cup \{t_f\})}_{\leq 0 \text{ due to the definition of } T_{k+1}, t_k^*} \\
&\leq \lim_{k\to\infty} \varphi(\bar{x}, T_k) - \varphi(x_k, T_{k+1}) \\
&= \lim_{k\to\infty} \underbrace{\varphi(\bar{x}, T_k) - \varphi(\bar{x}, T_{k+1})}_{\to 0 \text{ due to Proposition 2.1.14}} + \underbrace{\varphi(\bar{x}, T_{k+1}) - \varphi(x_k, T_{k+1})}_{\to 0 \text{ due to Proposition 2.1.18}} \\
&= 0
\end{aligned}$$

This contradicts the existence of such a $t_f \in T$. Consequently, the candidate-condition is fulfilled for $\bar{x} \in X$ or equivalently $\bar{x} \in C(S)$. Using the convergence Theorem 2.2.3 the claim holds. □

Additionally to the convergence of the iterates, the HLAD approach has another property that the uniform and LLAD do not have as they discretize the whole set T. The HLAD concentrates on the scenarios that guarantee feasibility of the accumulation points of $(x_k)_{k\in\mathbb{N}}$. The next lemma states that this leads to a monotone behavior in the probability evaluations. The behavior is noticeable, when the probust constraint is independent from $x \in X$. Therefore, the next lemma focuses on approximating probust terms with the HLAD.

Lemma 2.3.4 (Monotonicity of HLAD)
Assuming that in a probust term φ the inner function g is continuous w.r.t. (ξ, t) and Assumption 2.1.6 holds, then we can conclude that the subset sequence $(T_k)_{k\in\mathbb{N}}$ defined by HLAD (for probust terms) satisfies for any $k \in \mathbb{N}$

$$\varphi(T_k) - \varphi(T_{k+1}) \geq \varphi(T_{k+1}) - \varphi(T_{k+2}).$$

Proof. We show the claim directly using well-known set theoretical operations:
By Assumption 2.1.6 we know that $\varphi(S \cup \{t\})$ is a continuous function w.r.t. $t \in T$ for all $S \subseteq T$. Because $T \subseteq \mathbb{R}^q$ is compact, the new discretization point $t_k^* \in T$ defined by the HLAD is well-defined due to Weierstrass' theorem for all $k \in \mathbb{N}$.
Next we show that

$$\varphi(T_{k+2}) \geq 2\varphi(T_{k+1}) - \varphi(T_k). \tag{2.7}$$

By definition of $\varphi(S), S \subseteq T$ and t_k^* we know for any $k \in \mathbb{N}$:

$$\begin{aligned}
\varphi(T_{k+2}) &= \mathbb{P}\left(\bigcap_{t\in T_{k+2}} \Omega(t)\right) \\
&= \mathbb{P}\left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_k^*) \cap \Omega(t_{k+1}^*)\right) \\
&= \mathbb{P}\left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_k^*) \cap \bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_{k+1}^*)\right) \\
&= \mathbb{P}\left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_k^*)\right) + \mathbb{P}\left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_{k+1}^*)\right) \\
&\quad -\mathbb{P}\left(\left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_k^*)\right) \cup \left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_{k+1}^*)\right)\right) \\
&= \varphi(T_{k+1}) + \varphi(T_k \cup \{t_{k+1}^*\}) - \mathbb{P}\left(\left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_k^*)\right) \cup \left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_{k+1}^*)\right)\right) \\
&\geq \varphi(T_{k+1}) + \varphi(T_{k+1}) - \mathbb{P}\Bigg(\underbrace{\left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_k^*)\right) \cup \left(\bigcap_{t\in T_k} \Omega(t) \cap \Omega(t_{k+1}^*)\right)}_{\subseteq \bigcap_{t\in T_k} \Omega(t)}\Bigg) \\
&\geq \varphi(T_{k+1}) + \varphi(T_{k+1}) - \mathbb{P}\left(\bigcap_{t\in T_k} \Omega(t)\right) \\
&= 2\varphi(T_{k+1}) - \varphi(T_k)
\end{aligned}$$

As inequality (2.7) is equivalent to the claim, the claim holds. □

This means when using HLAD as a discretization method we can use the following numerical stopping criterion for a fixed precision $\epsilon > 0$ which heuristically estimates the feasibility of the current iterate x_k with $k \in \mathbb{N}$

$$\varphi(x_k, T_k) - \varphi(x_k, T_{k+1}) < \epsilon.$$

Multi-index high-level adaptive discretization approach (MIHLAD)

Last, but not least, we specify the subset scheme $\Phi = (\Phi_k)_{k\in\mathbb{N}}$ in the probust discretization algorithm introduced in Section 2.2 as

$$\Phi_k : X \times 2^T \to 2^T, (x, S) \mapsto S_k^*,$$

where $S_k^* := \{t_1^*, ..., t_{n_k}^*\}$ with $(t_1^*, ..., t_{n_k}^*) \in \arg\min_{(t_1,...,t_{n_k})\in T^{n_k}} \varphi(x, \{t_1, ..., t_{n_k}\})$ is the n_k-tuple of worst-scenarios considering the probability of $x \in X$ to be feasible with $k \in \mathbb{N}$ and $(n_k)_{k\in\mathbb{N}}$ is a sequence of natural numbers satisfying $n_k \geq k$ for all $k \in \mathbb{N}$. By definition this scheme is a discretization scheme. In comparison to the HLAD we do not add worst-case discretization points to a given subset S. Instead, we reconstruct the whole discretization in each iteration $k \in \mathbb{N}$. Consequently, MIHLAD does not have to be an increasing discretization scheme.

Lemma 2.3.5 (Convergence of MIHLAD)
Choosing Φ as the multi-index high-level adaptive discretization approach, any accumulation point $\bar{x}$ of $(x_k)_{k\in\mathbb{N}}$ converges to an minimizer of the original probust optimization problem.

Proof. We prove this result by using the reduction Theorem 2.2.6 and a reduction to the uniform discretization scheme with $d_k = \frac{1}{2^k}$ for all $k \in \mathbb{N}$:
Due to Assumption 2.1.6 and the compactness of $T \subseteq \mathbb{R}^q$ the new discretization points are well-defined for all $k \in \mathbb{N}$. As $X \subseteq \mathbb{R}^n, n \in \mathbb{N}$ is compact, the sequence $(x_k)_{k\in\mathbb{N}}$ has at least one accumulation point. Without loss of generality we assume that the whole sequence has just one accumulation point. Otherwise, we switch notations to the subsequence.
Now we fix this accumulation point $\bar{x} \in X$ and look at the sequence $(T_k)_{k\in\mathbb{N}}$ generated by the uniform discretization scheme. We have already shown that $S_1 := \lim_{k\to\infty} T_k$ guarantees $x \in C(S_1)$ for all $x \in X$ in Lemma 2.3.1. This also implies $\bar{x} \in C(S_1)$.
Then we can choose for any fixed $j \in \mathbb{N}$ the corresponding $k_j := |T_j| < \infty$ number of points in T_j. By definition of Φ_2 as the MIHLAD, we know that we choose in the k_jth iteration a subset $S_{k_j}^* \subseteq T$ with at least $|T_j|$ elements that fulfills by definition:

$$\varphi(x_{k_j}, S_{k_j}) = \min_{S\subseteq T, |S|=|S_{k_j}|} \varphi(x_{k_j}, S) \leq \varphi(x_{k_j}, T_j) \ \forall j \in \mathbb{N}$$

As $j \in \mathbb{N}$ was fixed arbitrarily, Theorem 2.2.6 guarantees us the existence of a subsequence of $(x_k)_{k\to\infty}$ such that:

$$\lim_{j\to\infty} \varphi(x_{k_j}, S_{k_j}) = \varphi(\bar{x}, T).$$

Since we assumed that $(x_k)_{k\in\mathbb{N}}$ is convergent, we know that $\bar{x}$ is a minimizer of the original probust optimization problem. □

Because the MIHLAD provides a more precise approximations of $\varphi(x_k, T)$ than HLAD given a fixed number of discretization points, we can also fix an arbitrary $\epsilon > 0$ and choose

$$\varphi(x_k, T_k) - \varphi(x_k, T_{k+1}) < \epsilon$$

as a stopping criterion for the probust subset algorithm.

Summary

In this chapter, we analyzed probust terms and found characterizations of how a subset $S \subseteq T$ can represent the whole set T in means of a probust evaluation. With this characterization in form of the candidate-condition in mind, we defined an iterative algorithm to solve standard probust optimization problems. To show that a given subset scheme Φ leads to the convergence of the corresponding iterates towards a minimizer of the original standard probust optimization problem, we gave two alternatives. On the one hand, we can directly show that any accumulation point of the iterates satisfies the candidate-condition. On the other hand, we can show that the considered subset scheme is more accurate than a reference subset scheme whose convergence is already guaranteed.

In the end, we showed that some prominent discretization schemes like an uniform discretization or the HLAD - as a probust version of the adaptive discretization scheme from Blankenship and Falk - work out in the probust setting.

One structural property of this subset approach is that we generate iterates $x_k, k \in \mathbb{N}$ that are in general infeasible w.r.t. the original probust optimization problem. Furthermore, the stopping criterion for the iterative probust subset algorithm is not clearly defined in general. We can use some minimal grid size in the uniform discretization approach or the condition $\varphi(x_k, T_k) - \varphi(x_k, T_{k+1}) < \epsilon$ for some precision $\epsilon > 0$ in the HLAD recalling Lemma 2.3.4, but these are just heuristics so far.

Our next goal is to define feasible iterates for probust optimization problem such that we can define upper bounds for the optimal objective value of a standard probust optimization problem next to the lower bounds that are defined by the iterates of the probust subset algorithm.

3 Set-approximation schemes

In the last chapter, we developed an iterative solution method for standard probust optimization problems based on subsets S of the set of scenarios T. As we do not consider the constraints indexed by $t \in T \setminus S$ in this approach, the generated iterates are in general infeasible w.r.t. the original standard probust optimization problem. Furthermore, we did not specify the stopping criterion for the iterative probust subset algorithm 3.
In this chapter, we introduce a second numerical approach to solve standard probust optimization problems that is based on approximating the set of feasible realizations $\Omega(x^*)$ of an optimal solution x^* of a standard probust optimization problem.
Over the course of this chapter, we see how the structure of the inner function g influences the structure of the set of feasible realizations and how we can use this structure to generate decisions that are feasible for the corresponding probust optimization problem.
We end this chapter with an example of how we use the set-approximation approach with a probust optimization problem.
As the set-approximation approach is not specifically designed for probust optimization problems, but for chance constrained optimization problems in general as introduced in Section 1.1, we mainly focus on this problem class. We then see that the results from chance constrained optimization problems can easily be transferred to probust optimization problems. The additional structure of probust optimization problems can even be used to specify special set-approximation approaches. Here, we are mainly inspired by the approximation of convex sets by linear inequalities.

3.1 Probability approximation by set-approximation

We start this chapter by considering the problem of approximating the probability of a given measurable set Ω by a family of measurable sets $(\Omega_\delta)_{\delta \in \Delta}$. We discuss a solution approach where we guarantee subset relations of Ω and the elements $\Omega_\delta, \delta \in \Delta$.
We verify that this relation leads to upper and lower bounds to the probability of the set Ω. We also define a condition such that these bounds are sharp.
After giving examples which family of sets satisfy this condition, we concentrate on measurable sets described by inequality constraints. In this context, we ask how we have to approximate these inequalities to guarantee a good approximation of the corresponding measurable sets.

Approximation by a family of sets

Given a probability space $(\Xi, \mathcal{A}, \mathbb{P})$, we focus on approximating the probability of a measurable set $\Omega \in \mathcal{A}$ by other measurable sets.
Here, we are looking for a set $\tilde{\Omega} \in \mathcal{A}$ which minimizes its probability of the symmetric difference with Ω that can be defined as

$$d_{\mathbb{P}}(\Omega, \tilde{\Omega}) := \mathbb{P}(\Omega \cap \tilde{\Omega}^C) + \mathbb{P}(\Omega^C \cap \tilde{\Omega}).$$

As this function is hard to evaluate in general, we are interested in reformulations to be able to find a best approximation $\tilde{\Omega}$ in the family of measurable sets $(\Omega_\delta)_{\delta \in \Delta}$.

Definition 3.1.1 (Set-approximation of a probability)
Let $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $\Omega \in \mathcal{A}$ and $(\Omega_\delta)_{\delta \in \Delta} \subseteq \mathcal{A}$ be a family of measurable sets with index set Δ.
We call the optimization problem

$$\inf_{\delta \in \Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta)$$

the set-approximation problem of Ω given $(\Omega_\delta)_{\delta \in \Delta}$,

$$\sup_{\delta \in \Delta} \mathbb{P}(\Omega_\delta) \ \textit{s.t.}\ \Omega_\delta \subseteq \Omega$$

the inner set-approximation problem of Ω given $(\Omega_\delta)_{\delta \in \Delta}$ and

$$\inf_{\delta \in \Delta} \mathbb{P}(\Omega_\delta) \ \textit{s.t.}\ \Omega \subseteq \Omega_\delta$$

the outer set-approximation problem of Ω given $(\Omega_\delta)_{\delta \in \Delta}$.
Here Δ is called the design space, $\delta \in \Delta$ is called a design parameter, Ω_δ is called a design and $\mathcal{D} = \bigcup_{\delta \in \Delta} \Omega_\delta$ is called set of designs.

Because Ω and Ω_δ are measurable for all $\delta \in \Delta$, we know that $d_{\mathbb{P}}(\Omega, \Omega_\delta) \in [0, 1]$ is well-defined for all $\delta \in \Delta$. To show the connection between the set-approximation problem and the inner/outer set-approximation problem of a $\Omega \in \mathcal{A}$, we need the following proposition:

Proposition 3.1.2 (Reformulation of set-approximation problem)
Let $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $\Omega \in \mathcal{A}$ and $(\Omega_\delta)_{\delta \in \Delta} \subseteq \mathcal{A}$ be a family of measurable sets. Then

(i) $\inf_{\delta \in \Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta) = \mathbb{P}(\Omega) - \sup_{\delta \in \Delta} \mathbb{P}(\Omega_\delta)$ *if $\Omega_\delta \subseteq \Omega$ holds for all $\delta \in \Delta$.*

(ii) $\inf_{\delta \in \Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta) = \inf_{\delta \in \Delta} \mathbb{P}(\Omega_\delta) - \mathbb{P}(\Omega)$ *if $\Omega \subseteq \Omega_\delta$ holds for all $\delta \in \Delta$.*

Proof. We show both reformulation using well-known operations from set and probability theory:
Proof of (i): Assume that $\Omega_\delta \subseteq \Omega$ holds for all $\delta \in \Delta$. Then we know that

$$\Omega_\delta \cap \Omega^C \subseteq \Omega \cap \Omega^C = \emptyset$$

holds for all $\delta \in \Delta$. Consequently, we can write

$$\begin{aligned}\inf_{\delta\in\Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta) &= \inf_{\delta\in\Delta} \mathbb{P}(\Omega \cap \Omega_\delta^C) + \mathbb{P}(\Omega^C \cap \Omega_\delta)\\ &= \inf_{\delta\in\Delta} \mathbb{P}(\Omega \cap \Omega_\delta^C) + 0\\ &= \inf_{\delta\in\Delta} \big(1 - \mathbb{P}(\Omega_\delta \cup \Omega^C)\big)\\ &= 1 - \sup_{\delta\in\Delta} \mathbb{P}(\Omega_\delta \cup \Omega^C)\\ &= 1 - \left(\sup_{\delta\in\Delta} \mathbb{P}(\Omega_\delta) + \mathbb{P}(\Omega^C)\right)\\ &= 1 - \mathbb{P}(\Omega^C) - \sup_{\delta\in\Delta} \mathbb{P}(\Omega_\delta)\\ &= \mathbb{P}(\Omega) - \sup_{\delta\in\Delta} \mathbb{P}(\Omega_\delta),\end{aligned}$$

where we used that $\Omega_\delta \cup \Omega^C$ is a union of disjoint sets as we assumed $\Omega_\delta \subseteq \Omega$. This shows the first claim.
Proof of (ii): Assume that $\Omega \subseteq \Omega_\delta$ holds for all $\delta \in \Delta$. Then we know that

$$\begin{aligned}\mathbb{P}(\Omega_\delta) &= \mathbb{P}(\Omega_\delta \cap \Omega) + \mathbb{P}(\Omega_\delta \cap \Omega^C)\\ &= \mathbb{P}(\Omega) + \mathbb{P}(\Omega_\delta \cap \Omega^C)\end{aligned}$$

holds for all $\delta \in \Delta$. Consequently, we can write

$$\begin{aligned}\inf_{\delta\in\Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta) &= \inf_{\delta\in\Delta} \mathbb{P}(\Omega \cap \Omega_\delta^C) + \mathbb{P}(\Omega^C \cap \Omega_\delta)\\ &= \inf_{\delta\in\Delta} 0 + \mathbb{P}(\Omega^C \cap \Omega_\delta)\\ &= \inf_{\delta\in\Delta} \mathbb{P}(\Omega_\delta) - \mathbb{P}(\Omega).\end{aligned}$$

This shows that the second claim holds. □

While the function $f(\delta) := d_{\mathbb{P}}(\Omega, \Omega_\delta), \delta \in \Delta$ seems hard to analyse because we do not know how correspondences handle relative complements, the last Proposition allows us to work with $\tilde{f}(\delta) = \mathbb{P}(\Omega_\delta)$ instead. By Lemma 1.2.17 we know, that an outer semi-continuous correspondence $\Omega : \Delta \rightrightarrows \Xi$ is enough to ensure the upper semi-continuity of $\tilde{f}$ and therefore guarantee a well-defined maximum $\max_{\delta\in\Delta} \mathbb{P}(\Omega_\delta)$ if the design space Δ is compact.
That under such assumptions the inner and outer set-approximation problems really define inner and outer approximations of the probability is shown by the next proposition:

Proposition 3.1.3 (Approximation via inner/outer set-approximation)
Given a probability space $(\Xi, \mathcal{A}, \mathbb{P})$ whose probability distribution has a Lebesgue-density, a $\Omega \in \mathcal{A}$, a compact set Δ and a family of sets $(\Omega_\delta)_{\delta\in\Delta} \subseteq \mathcal{A}$.

(*i*) *Assume further that the correspondence $\Gamma : \Delta \rightrightarrows \Xi, \delta \mapsto \Omega_\delta$ is outer semi-continuous and that $\Omega_\delta \subseteq \Omega$ holds for all $\delta \in \Delta$. Then the optimal value $\underline{p} := \max_{\delta\in\Delta} \mathbb{P}(\Omega_\delta)$ satisfies $\underline{p} \leq \mathbb{P}(\Omega)$.*

(*ii*) *Alternatively, assume further that the correspondence $\Gamma : \Delta \rightrightarrows \Xi, \delta \mapsto \Omega_\delta$ is inner semi-continuous and that $\Omega \subseteq \Omega_\delta$ holds for all $\delta \in \Delta$. Then the optimal value $\overline{p} := \min_{\delta\in\Delta} \mathbb{P}(\Omega_\delta)$ satisfies $\mathbb{P}(\Omega) \leq \overline{p}$.*

Proof. We proof both claims in two steps. First, we show that the inner/outer set-approximation problem has a well-defined maximizer/minimizer $\delta^* \in \Delta$. Second, we use the subset relations to guarantee $\underline{p} \leq \mathbb{P}(\Omega) \leq \overline{p}$:

Proof of (*i*): We use Lemma 1.2.17 (*i*) to conclude that $f : \Delta \to [0, 1], \delta \mapsto \mathbb{P}(\Gamma(\delta))$ is an upper semi-continuous function. As Δ is compact, we know that the optimization problem $\max_{\delta\in\Delta} \mathbb{P}(\Omega_\delta)$ has at least one well-defined maximizer $\delta^* \in \Delta$.
Let us fix an arbitrary design parameter $\delta \in \Delta$. Because we assumed $\Omega_\delta \subseteq \Omega$ for all $\delta \in \Delta$, we know that $\mathbb{P}(\Omega_\delta) \leq \mathbb{P}(\Omega)$ for all $\delta \in \Delta$. This is especially true for all maximizers and therefore $\underline{p} \leq \mathbb{P}(\Omega)$ holds.

Proof of (*ii*): We use Lemma 1.2.17 (*i*) to conclude that $f : \Delta \to [0, 1], \delta \mapsto \mathbb{P}(\Gamma(\delta))$ is a lower semi-continuous function. As Δ is compact, we know that the optimization problem $\min_{\delta\in\Delta} \mathbb{P}(\Omega_\delta)$ has at least one well-defined minimizer $\delta^* \in \Delta$.
Let us fix an arbitrary design parameter $\delta \in \Delta$. Because we assumed $\Omega \subseteq \Omega_\delta$ for all $\delta \in \Delta$, we know that $\mathbb{P}(\Omega) \leq \mathbb{P}(\Omega_\delta)$ for all $\delta \in \Delta$. This is especially true for all minimizers and therefore $\mathbb{P}(\Omega) \leq \overline{p}$ holds. □

To use the first part of this proposition, we have to guarantee that $\Omega_\delta \subseteq \Omega$ holds for all $\delta \in \Delta$. A simple way to do so is to fix an arbitrary subset $(\Omega_\delta)_{\delta\in\Delta} \subseteq \mathcal{A}$ and then look at tits intersection with Ω. A family of sets $(\hat{\Omega}_\delta)_{\delta\in\Delta}$ constructed that way is also measurable as the intersection of two measurable sets and satisfies $\hat{\Omega}_\delta \subseteq \Omega$ for all $\delta \in \Delta$.
To use the second part of the proposition, we can construct a new family of sets by $\hat{\Omega}_\delta := \Omega_\delta \cup \Omega$ for all $\delta \in \Delta$.
As we can use Proposition 3.1.3 to define upper and lower bounds for $\mathbb{P}(\Omega)$, we want to know under which condition we can ensure $\underline{p} = \mathbb{P}(\Omega) = \overline{p}$.

Definition 3.1.4 (Fitness-condition)
Let $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space and $\Omega \in \mathcal{A}, \mathcal{D} = (\Omega_\delta)_{\delta\in\Delta} \subseteq \mathcal{A}$ for some index set Δ. We say that $\mathcal{D}$ satisfies the fitness-condition, if

$$\inf_{\delta\in\Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta) = 0.$$

This condition implies the required equality as we see in the next proposition:

Proposition 3.1.5 (Sufficiency of fitness-condition)
Given a measurable space $(\Xi, \mathcal{A}, \mathbb{P})$ whose probability distribution has a Lebesgue-density, a measurable set $\Omega \in \mathcal{A}$, a compact set Δ and a family of sets $(\Omega_\delta)_{\delta \in \Delta} \subseteq \mathcal{A}$ that fulfills the fitness-condition.

(i) *Assume further that the correspondence $\Gamma : \Delta \rightrightarrows \Xi, \delta \mapsto \Omega_\delta$ is upper hemi-continuous and that $\Omega_\delta \subseteq \Omega$ holds for all $\delta \in \Delta$. Then the maximal lower bound $\underline{p} := \max_{\delta \in \Delta} \mathbb{P}(\Omega_\delta)$ satisfies $\underline{p} = \mathbb{P}(\Omega)$.*

(ii) *Alternatively, assume further that the correspondence $\Gamma : \Delta \rightrightarrows \Xi, \delta \mapsto \Omega_\delta$ is lower hemi-continuous and that $\Omega \subseteq \Omega_\delta$ holds for all $\delta \in \Delta$. Then the minimal upper bound $\overline{p} := \min_{\delta \in \Delta} \mathbb{P}(\Omega_\delta)$ satisfies $\mathbb{P}(\Omega) = \overline{p}$.*

Proof. We proof the statements using the fitness-condition, Proposition 3.1.2 and Proposition 3.1.3:
Proof of (i): The fitness-condition guarantees us, that

$$\inf_{\delta \in \Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta) = 0.$$

By Proposition 3.1.2 (i) we know that

$$\inf_{\delta \in \Delta} d_{\mathbb{P}}(\Omega, \Omega_\delta) = \mathbb{P}(\Omega) - \sup_{\delta \in \Delta} \mathbb{P}(\Omega_\delta) = 0.$$

We know by Proposition 3.1.3 (i) that $\max_{\delta \in \Delta} \mathbb{P}(\Omega_\delta) \leq \mathbb{P}(\Omega)$ and that its maximum will be attained. Consequently, the last equality gives us

$$\mathbb{P}(\Omega) = \mathbb{P}(\Omega_{\delta^*}) = \underline{p}.$$

That shows the first claim.
Proof of (ii): Analogously, we can use Proposition 3.1.2 (ii) and Proposition 3.1.3 (ii) to show

$$\mathbb{P}(\Omega) = \mathbb{P}(\Omega_{\delta^*}) = \overline{p},$$

what proves the second claim. □

To understand the fitness-condition better, we consider the following examples:

Example 3.1.6 (Family of sets fulfilling the fitness-condition)
(i) *Let $([-1, 2], \mathcal{B}, \mathbb{P})$ be a probability space, where $\mathbb{P}$ is induced by $Z \sim \mathcal{U}([0, 1])$. Let $\Omega = [-0.5, 1] \in \mathcal{B}$ and define $\Delta := [0, 1]$, $\Omega_\delta := [0, \delta]$. Then all defined sets $\Omega_\delta \in \mathcal{B}$ are measurable and the family $\mathcal{D} = (\Omega_\delta)_{\delta \in \Delta}$ fulfills the fitness-condition for Ω because the choice $\delta^* := 1.5$ leads to*

$$d_{\mathbb{P}}(\Omega, \Omega_{1.5}) = \mathbb{P}([-0.5, 0) \cup (1, 1.5]) = 0.$$

(ii) Let $(\Xi, \mathcal{B}, \mathbb{P})$ be a probability space with $\Xi \subseteq \mathbb{R}^m$ such that $\mathbb{P}$ has a bounded density $\rho \in L^\infty(\Xi, \mathbb{R})$. Let $\Omega \in \mathcal{A}$ be a compact, convex set, define the design space the sets of polyhedra described by a natural number of linear inequality constraints by

$$\Delta := \bigcup_{k \in \mathbb{N}} \Delta_k, \Delta_k = \{(A, b) \in \mathbb{R}^{m \times k} \times \mathbb{R}^k\},$$
$$\Omega_\delta := \{\xi \in \Xi \mid A\xi \leq b\} \; \forall \delta = (A, b) \in \Delta.$$

Then all defined sets $\Omega_\delta \in \mathcal{B}$ are measurable and the family $\mathcal{D} = (\Omega_\delta)_{\delta \in \Delta}$ fulfills the fitness-condition for Ω. We know by Theorem 4.1 of a paper by Dudley [27] that we can approximate any compact, convex set by polyhedrons arbitrarily good w.r.t. the Hausdorff-metric and the Lebesgue-measure of their symmetric difference. Using

$$\mathbb{P}(B) = \int_B 1 d\mathbb{P}(\xi) = \int_B \rho(\xi) d\lambda(\xi) \leq ||\rho||_{L^\infty(\Xi,\mathbb{R})} \lambda(B)$$

with any measurable set $B \in \mathcal{B}$ this leads to the estimation

$$\lim_{k\to\infty} \inf_{\delta \in \Delta_k} d_{\mathbb{P}}(\Omega, \Omega_\delta) \leq \lim_{k\to\infty} ||\rho||_{L^\infty(\Xi,\mathbb{R})} \lambda\left(B_k\right) \leq \lim_{k\to\infty} ||\rho||_{L^\infty(\Xi,\mathbb{R})} \frac{c(\Omega)}{k^{\frac{2}{m-1}}} = 0,$$

where $B_k = (\Omega \cap \Omega^C_{\delta^*_k}) \cup (\Omega^C \cap \Omega_{\delta^*_k})$. Consequently, the fitness-condition is fulfilled.

After considering set-approximation problems with a family of measurable sets $(\Omega_\delta)_{\delta\in\Delta}$, we are interested in how we can use inequality constraints to describe such sets.

Approximation by a family of functions

So far, we have discussed sufficient conditions for a given family of sets to approximate $\mathbb{P}(\Omega)$ arbitrarily good. As we are free to chose any family of sets $(\Omega_\delta)_{\delta\in\Delta} \subseteq \mathcal{A}$, we are now interested how to specify it.
We assume that Ω can be described by an inequality constraint. This means that there exists a $\mathcal{A}$-measurable function $g : \Xi \to \mathbb{R}$ such that $\Omega = \{\xi \in \Xi \mid g(\xi) \leq 0\}$.
Furthermore, we decide to approximate this Ω by measurable sets $(\Omega_\delta)_{\delta\in\Delta}$ that can also be described by functions. We assume that for all $\delta \in \Delta$ there exists $\mathcal{A}$-measurable functions $g_\delta : \Xi \to \mathbb{R}$ such that $\Omega_\delta = \{\xi \in \Xi \mid g_\delta(\xi) \leq 0\}$.
With this representation, we can rewrite the inner and outer set-approximation problem given $\Omega \in \mathcal{A}$ as

$$\sup_{\delta\in\Delta} \mathbb{P}(\Omega_\delta) \text{ s.t. } \Omega_\delta \subseteq \Omega$$
$$\Leftrightarrow \sup_{\delta\in\Delta} \mathbb{P}(\Omega_\delta) \text{ s.t. } g(\xi) \leq 0 \; \forall \xi \in \Omega_\delta$$

and

$$\inf_{\delta\in\Delta} \mathbb{P}(\Omega_\delta) \text{ s.t. } \Omega \subseteq \Omega_\delta$$
$$\Leftrightarrow \inf_{\delta\in\Delta} \mathbb{P}(\Omega_\delta) \text{ s.t. } g_\delta(\xi) \leq 0 \; \forall \xi \in \Omega.$$

If we consider a finite-dimensional search space $\Delta \subseteq \mathbb{R}^d$, the reformulation of the inner approximation problem is a generalized semi-infinite optimization problem, while the reformulation of the the outer approximation problem is a standard semi-infinite optimization problem.

We are interested in how the structure of the function g influences the structure of the set

$$\Omega(g) = \{\xi \in \Xi \mid g(\xi) \leq 0\}.$$

Therefore, we use the following proposition:

Proposition 3.1.7 (Structure of set of feasible realizations)
Let $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space and $g : \Xi \to \mathbb{R}$ satisfy $\Omega(g) \in \mathcal{A}$. Then the following holds:
(i) If g is lower semi-continuous, then $\Omega(g)$ is closed.
(ii) If g is quasi-convex, then $\Omega(g)$ is convex.
(iii) If g is lower semi-continuous as well as coercive, then $\Omega(g)$ is bounded.
(iv) If $\Xi \subseteq \mathbb{R} \cup \{\pm\infty\}$ and g is monotonically increasing as well as continuous, then there exists a $M \in \bar{\mathbb{R}}$ such that $\Omega(g) = (-\infty, M] \cap \Xi$, where in this context $(-\infty, -\infty] = \emptyset$ and $(-\infty, \infty] = \mathbb{R}$.

Proof. We show the claims individually using well-known arguments from analysis and set theory:
Proof of (i): We show that for any convergent sequence $(\xi_k)_{k\in\mathbb{N}} \subseteq \Omega(g)$ its limit is again in $\Omega(g)$:
Consider an arbitrary, but fixed convergent sequence $(\xi_k)_{k\in\mathbb{N}} \subseteq \Omega(g)$. By definition of Ω we know that $g(\xi_k) \leq 0$ for all $k \in \mathbb{N}$. As g was assumed to be lower semi-continuous, we know that at the limit point $\bar{\xi} \in \Xi$ of $(\xi_k)_{k\in\mathbb{N}}$ holds

$$g(\bar{\xi}) \leq \lim_{k\to\infty} g(\xi_k) \leq 0.$$

This ensures $\bar{\xi} \in \Omega(g)$. Because $(\xi_k)_{k\in\mathbb{N}}$ was chosen arbitrarily, we know by definition that $\Omega(g)$ is closed what shows the first claim.

Proof of (ii): We show that for any elements $\xi_1, \xi_2 \in \Omega(g), \lambda \in [0,1]$ their convex combination $\xi(\lambda) = \lambda\xi_1 + (1-\lambda)\xi_2$ is also an element of $\Omega(g)$:
Consider arbitrary, but fixed elements $\xi_1, \xi_2 \in \Omega(g), \lambda \in [0,1]$. By definition of Ω we know that $g(\xi_i) \leq 0$ for $i = 1, 2$. As g was assumed to be quasi-convex, we know that for any $\lambda \in [0,1]$ it holds

$$g(\xi(\lambda)) \leq \max\{g(\xi_1), g(\xi_2)\} \leq 0.$$

This implies $\xi(\lambda) \in \Omega(g)$. Since $\lambda \in [0,1], \xi_1, \xi_2 \in \Omega(g)$ were chosen arbitrarily, we know by definition that $\Omega(g)$ is convex what shows the second claim.

Proof of (iii): Assuming that g is coercive, we argue indirectly:
Let us assume that $g : \Xi \to \mathbb{R}$ is coercive, but $\Omega(g)$ is unbounded. Then there exists a sequence $(\xi_k)_{k\in\mathbb{N}} \subseteq \Omega(g)$ with $\lim_{k\to\infty} ||\xi_k|| = \infty$. As g was assumed to be coercive, we know that

$$\lim_{k\to\infty} g(\xi_k) = \infty.$$

As g is lower semi-continuous there exists a $N \in \mathbb{N}$ such that $g_k(\xi) > 0$ holds for all $k \geq N$. This contradicts $\xi_k \in \Omega(g)$ for all $k \in \mathbb{N}$.
Consequently, the assumption that $\Omega(g)$ is unbounded is wrong what shows the third claim.

Proof of (iv): We show that claim directly by a case distinction and the monotonicity:
1. Case: If $\Omega(g) = \emptyset$, we set $M = -\infty$ and the claim holds.
2. Case: If $\Omega(g) \neq \emptyset$ is bounded from above. Then their exists some $M = \sup_{\xi\in\Omega(g)} \xi$. As g is continuous and $\Omega(g)$ is bounded from above, we know that $M \in \Omega(g)$. Consequently, we know that $\xi \leq M$ for all $\xi \in \Omega(g)$ and therefore $\Omega(g) \subseteq (-\infty, M] \cap \Xi$. By the monotonicity of g, we know that $\Omega(g) = (-\infty, M] \cap \Xi$.
3. Case: If $\Omega(g)$ is unbounded from above, we know that for all $N \in \mathbb{N}$ there exists a $\xi_N \in \Omega(g)$ such that $N \leq \xi_N$. As g is monotonically increasing, we know that $\Omega(g)$ is a superset of $(-\infty, N] \cap \Xi$ for all $N \in \mathbb{N}$ and consequently $M = \infty$ fulfills the claim, what shows the forth and last claim. □

In conclusion, it appears wise to approximate g by a function $\tilde{g}$ that has the same structure. On the contrary, the following example shows that a convergent uniform approximation does not lead to the convergence of the corresponding probabilities in general and vice versa.

Example 3.1.8 (Approximating g by $\tilde{g}$)
(i) Let us consider any probability space $(\Xi, \mathcal{A}, \mathbb{P})$ and any inner function $g : \Xi \to \mathbb{R}$ that satisfies $\Omega(g) \in \mathcal{A}$. Then we can define

$$\tilde{g}(\xi) := s(\xi)g(\xi)$$

with an arbitrary function $s : \Xi \to \mathbb{R}_{>0}$. As we do not switch the sign between $\tilde{g}(\xi)$ and $g(\xi)$ for any $\xi \in \Xi$ by this multiplication, we know that $\Omega(g) = \Omega(\tilde{g})$ and $\varphi(g) = \varphi(\tilde{g})$.
(ii) Consider the probability space $([0,1], \mathcal{B}, \mathbb{P})$, where $\mathbb{P}$ is the probability measure induced by $Z \sim \mathcal{U}([0,1])$. Define $g : [0,1] \to \mathbb{R}, \xi \mapsto 0$ and $\tilde{g}_k : [0,1] \to \mathbb{R}, \xi \mapsto \frac{\xi}{k}$. By definition we know that $\tilde{g}_k \to g$ uniformly for $k \to \infty$ because

$$\lim_{k\to\infty} \max_{\xi\in[0,1]} |\tilde{g}_k(\xi) - g(\xi)| = \lim_{k\to\infty} \max_{\xi\in[0,1]} \left|\frac{\xi}{k}\right| = \lim_{k\to\infty} \frac{1}{k} = 0.$$

On the contrary, we know by definition of $g, \tilde{g}_k$ that $\Omega(g) = [0,1]$ and $\Omega(\tilde{g}_k) = \{0\}$ for all $k \in \mathbb{N}$ and therefore $\lim_{k\to\infty} \varphi(\tilde{g}_k) = 0 \neq 1 = \varphi(g)$.

The first example shows that we do not need a good approximation $\tilde{g}$ of g to guarantee $\varphi(g) = \varphi(\tilde{g})$. The second example shows that even a uniform approximation of the function g by $\tilde{g}$ can lead to the maximal possible error of 1. To improve the understanding of this convergence behavior, we consider the following statement:

Proposition 3.1.9 ($\tilde{g} \to g$ point-wise implies $\varphi(\tilde{g}) \to \varphi(g)$)
Let $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $g, \tilde{g}_k : \Xi \to \mathbb{R}$ be inner functions such that the corresponding sets $\Omega(g), \Omega(\tilde{g}_k)$ are elements of $\mathcal{A}$ for all $k \in \mathbb{N}$ and $\tilde{g}_k \to g$ point-wise.

(i) If $\tilde{g}_k(\xi) \leq g(\xi)$ for all $k \in \mathbb{N}, \xi \in \Xi$, then $\lim_{k\to\infty} \varphi(\tilde{g}_k) = \varphi(g)$.

(ii) If $\tilde{g}_k(\xi) \geq g(\xi)$ for all $k \in \mathbb{N}, \xi \in \Xi$, then $\limsup_{k\to\infty} \varphi(\tilde{g}_k) \leq \varphi(g)$.

(iii) If $\mathbb{P}(g(\xi) = 0) = 0$, then $\lim_{k\to\infty} \varphi(\tilde{g}_k) = \varphi(g)$.

Proof. To prove the first claim we show that $\chi_{\Omega(\tilde{g}_k)}$ converges point-wise towards $\chi_{\Omega(g)}$ indirectly and use the dominated convergence theorem afterwards:
Let us assume that there exists a realization $\xi_f \in \Xi$ such that $\chi_{\Omega(\tilde{g}_k)}(\xi_f)$ does not converge towards $\chi_{\Omega(g)}(\xi_f)$. This means that there exists an $\epsilon > 0$ such that for all $n \in \mathbb{N}$ there exists a $K_n \geq n$ such that

$$|\chi_{\Omega(\tilde{g}_{K_n})}(\xi_f) - \chi_{\Omega(g)}(\xi_f)| \geq \epsilon. \tag{3.1}$$

By definition of the characteristic function χ the realization ξ_f lies either within $\Omega(\tilde{g}_{K_n})$ or within $\Omega(g)$. Due to the assumption $\tilde{g}_k(\xi) \leq g(\xi)$ for all $\xi \in \Xi, k \in \mathbb{N}$, we know that $\tilde{g}_{k_N}(\xi_f) \leq 0$ and $g(\xi_f) > 0$ has to hold. As $\tilde{g}_k$ converges point-wise towards g, there exists a $N \in \mathbb{N}$ such that $|\tilde{g}_k(\xi_f) - g(\xi_f)| < \frac{g(\xi_f)}{2}$ for all $k \geq N$. Consequently, we can conclude for all $k \geq N$

$$0 \geq \tilde{g}_k(\xi_f) \geq g(\xi_f) - \frac{g(\xi_f)}{2} = \frac{g(\xi_f)}{2} > 0.$$

Fixing $n = N$, our assumption (3.1) is wrong and the $\chi_{\Omega(\tilde{g}_k)}$ converges point-wise towards $\chi_{\Omega(g)}$.
With that we can use the the dominated convergence theorem to state

$$\begin{aligned}
\lim_{k\to\infty} \varphi(\tilde{g}_k) &= \lim_{k\to\infty} \mathbb{P}(\Omega(\tilde{g}_k)) \\
&= \lim_{k\to\infty} \int_{\Xi} \chi_{\Omega(\tilde{g}_k)}(\xi) d\mathbb{P}(\xi) \\
&= \int_{\Xi} \lim_{k\to\infty} \chi_{\Omega(\tilde{g}_k)}(\xi) d\mathbb{P}(\xi) \\
&= \int_{\Xi} \chi_{\Omega(g)}(\xi) d\mathbb{P}(\xi) \\
&= \varphi(g),
\end{aligned}$$

what proves the first part of the claim.

To prove the second claim let us fix an arbitrary $k \in \mathbb{N}, \xi \in \Omega(\tilde{g}_k)$. As we assumed that $g(\xi) \leq \tilde{g}_k(\xi)$ for all $k \in \mathbb{N}, \xi \in \Xi$, we know that a realization $\xi \in \Omega(\tilde{g}_k)$ satisfies $\tilde{g}_k(\xi) \leq 0$ by definition. This implies $\xi \in \Omega(g)$ since

$$g(\xi) \leq \tilde{g}_k(\xi) \leq 0.$$

As $k \in \mathbb{N}, \xi \in \Omega(\tilde{g}_k)$ were chosen arbitrarily, we know that $\Omega(\tilde{g}_k) \subseteq \Omega(g)$ for all $k \in \mathbb{N}$. Consequently $\limsup_{k\to\infty} \varphi(\tilde{g}_k) \leq \varphi(g)$.

We prove the third claim by indirectly verifying that $\chi_{\Omega(\tilde{g}_k)}$ converges point-wise towards $\chi_{\Omega(g)}$ and using the dominated convergence theorem afterwards:
Let us assume that there exists a realization $\xi_f \in \Xi$ such that $\chi_{\Omega(\tilde{g}_k)}(\xi_f)$ does not converge towards $\chi_{\Omega(g)}(\xi_f)$. This means that there exists an $\epsilon > 0$ such that for all $n \in \mathbb{N}$ there exists a $K_n \geq n$ such that

$$|\chi_{\Omega(\tilde{g}_{K_n})}(\xi_f) - \chi_{\Omega(g)}(\xi_f)| \geq \epsilon. \tag{3.2}$$

By definition of the characteristic function χ the realization ξ_f is either an element of $\Omega(\tilde{g}_{K_n})$ or an element of $\Omega(g)$. Due to the assumption $\mathbb{P}(g(\xi) = 0) = 0$, we can assume without loss of generality that $g(\xi_f) \neq 0$. Due to the point-wise convergence, there exists some $N \in \mathbb{N}$ such that for all $k \geq N$ we can guarantee

$$|\tilde{g}_k(\xi_f) - g(\xi_f)| < \frac{|g(\xi_f)|}{2}.$$

Fixing $n = N$, we can argue like in (i) with a case distinction:
Case 1: If $\xi_f \in \Omega(\tilde{g}_{K_n}) \setminus \Omega(g)$, we know $g(\xi_f) > 0$. Now, the definition of $N \in \mathbb{N}$ implies for $k \geq N$

$$\tilde{g}_k(\xi_f) \geq g(\xi_f) - \frac{g(\xi_f)}{2} = \frac{g(\xi_f)}{2} > 0.$$

Fixing $n = N$ in assumption (3.2) this is a contradiction and $\chi_{\Omega(\tilde{g}_k)}$ converges point-wise towards $\chi_{\Omega(g)}$.
Case 2: If $\xi_f \in \Omega(g) \setminus \Omega(\tilde{g}_k)$, we know $g(\xi_f) < 0$. The definition of $N \in \mathbb{N}$ implies for $k \geq N$

$$\tilde{g}_k(\xi_f) \leq g(\xi_f) + \frac{-g(\xi_f)}{2} = \frac{g(\xi_f)}{2} < 0.$$

Fixing $n = N$ in assumption (3.2) this is a contradiction and $\chi_{\Omega(\tilde{g}_k)}$ converges point-wise towards $\chi_{\Omega(g)}$.

All together we know that $\chi_{\Omega(\tilde{g}_k)}$ converges $\mathbb{P}$-almost everywhere point-wise towards $\chi_{\Omega(g)}$ and we can use the the dominated convergence theorem to state:

$$\begin{aligned}\lim_{k\to\infty}\varphi(\tilde{g}_k) &= \lim_{k\to\infty}\mathbb{P}(\Omega(\tilde{g}_k)) \\ &= \lim_{k\to\infty}\int_\Xi \chi_{\Omega(\tilde{g}_k)}(\xi)d\mathbb{P}(\xi) \\ &= \int_\Xi \lim_{k\to\infty}\chi_{\Omega(\tilde{g}_k)}(\xi)d\mathbb{P}(\xi) \\ &= \int_\Xi \chi_{\Omega(g)}(\xi)d\mathbb{P}(\xi) \\ &= \varphi(g)\end{aligned}$$

This proves the third and last part of the claim. □

If we recall the probust subset algorithm from the last chapter to approximate the value of probust terms by skipping the decision calculation in every iteration, we can see that we used the functions $\sup_{t\in T_k} g(\cdot,t)$ to approximate $\max_{t\in T} g(\cdot,t)$. This setting fits to the assumptions of the last proposition part (i).

After this excursion into probability approximation by set-approximation, our next goal is to use the set-approximation approach on chance constrained optimization problems.

3.2 Approximation of chance constrained optimization problems

The main idea of this section is to use the introduced set-approximation approach for chance constrained optimization problems. Considering these problems, we see that a feasible decision $x \in X$ induces a measurable set $\Omega(x) \in \mathcal{A}$ such that $\mathbb{P}(\Omega(x)) \geq p$.
We will try to approximate this set $\Omega(x)$ by a family of measurable sets $(\Omega_\delta)_{\delta\in\Delta}$. After defining the corresponding inner set-approximation problem, we will proof that this problem really is an inner approximation of the original chance constrained optimization problem. Additionally, we comment on the convergence behavior of an iterative inner set-approximation algorithm.

Since we consider chance constrained optimization problems as introduced in Section 1.1, we are not interested in determining the exact probability $\mathbb{P}(\Omega)$ of some measurable set $\Omega \in \mathcal{A}$, but in underestimations of this probability.
To create an inner approximation, we therefore need to guarantee that $\mathbb{P}(\Omega_\delta) \geq p$ implies $\mathbb{P}(\Omega(x)) \geq p$. One way to do so is focusing on the constraint $\Omega_\delta \subseteq \Omega(x)$ for all $\delta \in \Delta$ as we have seen in Proposition 3.1.3 (i). As $\Omega_\delta \subseteq \Omega(x)$ might be to restrictive as a constraint considering an arbitrary $x \in X$, we allow ourselves to work with decision-dependent sets.

To be able to differentiate between the decision-dependent set $\Omega(x)$ and our designs, we switch the notation for the approximating sets from $\Omega(\delta)$ to $D(\delta)$ for the remainder of this chapter.

Definition 3.2.1 (Set-approximation problem of CC)
Let X, Δ be compact sets, $f : X \to \mathbb{R}$ be a continuous function, $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $g : X \times \Xi \to \mathbb{R}$ be a Caratheodory-function, $p \in [0,1]$ and $D : X \times \Delta \to \mathcal{A}$ a design function. We call

$$ISA_D : \min_{x \in X, \delta \in \Delta} f(x) \text{ s.t. } \mathbb{P}(D(x,\delta)) \geq p \tag{3.3}$$

$$\sup_{\xi \in D(x,\delta)} g(x,\xi) \leq 0 \tag{3.4}$$

the inner set-approximation problem of a chance constrained optimization problem CC.

To ensure that the feasible set of the inner set-approximation problem is nonempty, we need to introduce the following value which is closely related to the values defined in Equations (1.5) and (1.6):

$$\begin{aligned} p_{\max}(\mathcal{D}) = \max_{x \in X, \delta \in \Delta, p \in [0,1]} p \text{ s.t. } & \mathbb{P}(D(x,\delta)) \geq p \\ & \sup_{\xi \in D(x,\delta)} g(x,\xi) \leq 0, \\ \text{where } \mathcal{D} = \bigcup_{x \in X, \delta \in \Delta} & D(x,\delta). \end{aligned}$$

With this value we can state:

Proposition 3.2.2 (Inner approximation by inner set-approximation problem)
Let X, Δ be nonempty, compact sets, $f : X \to \mathbb{R}$ be a lower semi-continuous function, $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $g : X \times \Xi \to \mathbb{R}$ be a Caratheodory-function, $p \in [0,1]$ and $D : X \times \Delta \to \mathcal{A}$ a design function such that the correspondence $\Gamma : X \times \Delta \rightrightarrows \Xi, \delta \mapsto D(x,\delta)$ is continuous.
If $p \leq p_{\max}(\mathcal{D})$, then the inner set-approximation of the induced chance constrained problem is well-defined. Additionally, every feasible point of the inner set-approximation is also feasible w.r.t. the induced chance constrained problem.

Proof. First, we show the well-posedness of the problem by well-established theorems from the literature and afterwards show the feasible set inclusion directly:
We know by assumption that f is lower semi-continuous and that $X \times \Delta$ is compact as a finite Cartesian-product of compact sets. Consequently, we want to show that

$$\mathcal{F}_{D,p} := \{(x,\delta) \in X \times \Delta \mid \mathbb{P}(D(x,\delta)) \geq p \wedge \sup_{\xi \in D(x,\delta)} g(x,\xi) \leq 0\}$$

is a closed set. Please note that by the choice of the threshold $p \leq p_{\max}(\mathcal{D})$ we know that the feasible set $\mathcal{F}_{D,p}$ is nonempty.

By assumption Γ is continuous w.r.t. $(x, \delta) \in X \times \Delta$ and by Lemma 1.2.17 we know that the function $\mathbb{P}(D(\cdot))$ is upper semi-continuous w.r.t. $(x, \delta) \in X \times \Delta$.
By our assumptions Lemma 1.2.6 guarantees that the function $\sup_{\xi \in D(x,\delta)} g(x, \xi)$ is lower semi-continuous w.r.t $(x, \delta) \in X \times \Delta$ as well. Consequently, $\mathcal{F}_{D,p}$ is compact as the intersection of a compact set with a closed set and the optimization problem has a well-posed minimizer as f was assumed to be lower semi-continuous.

Next, we show the inclusion of feasible sets:
Let $(x, \delta) \in \mathcal{F}_{D,p}$ be a feasible point of the inner set-approximation problem. Then it holds that $\mathbb{P}(D(x, \delta)) \geq p$ and $g(x, \xi) \leq 0$ for all $\xi \in D(x, \delta)$. Consequently, we can ensure $D(x, \delta) \subseteq \Omega(x) = \{\xi \in \Xi \mid g(x, \xi) \leq 0\}$ what implies

$$\mathbb{P}(\Omega(x)) \geq \mathbb{P}(D(x, \delta)) \geq p.$$

This means that $x \in \mathcal{F}_p$ is feasible what shows the second part of the claim. □

Please note that the assumptions for this proposition are just slightly stronger than the assumptions we need to guarantee the existence of a minimizer in the chance constrained optimization context, namely we require Δ to be compact and D to be continuous as a correspondence.
If we follow the argumentation of the last chapter, we should try to generalize the fitness-condition 3.1.4 to ensure that the inner set-approximation converges to an optimal solution of the original chance constrained optimization problem.
That it is difficult to find conditions that guarantee the convergence of an inner approximation as we illustrate with the following counter example:

Example 3.2.3 (Counter example for convergence of inner set-approximation) *Chose $([0, 1], \mathcal{B}, \mathbb{P})$ as the probability space, where $\mathbb{P}$ is induced by the random variable $Z \sim \mathcal{U}([0, 1])$, $X := [-1, 1]$, $f(x) = x$, $g(x, \xi) := -x^2 - x - p + \xi$ and $p \in [0, 1]$. This data induces the following chance constrained optimization problem:*

$$\min_{x \in [-1,1]} x \text{ s.t. } \mathbb{P}(-x^2 - x - p + \xi \leq 0) \geq p$$

We can show that the decision $x^* := -1$ is feasible by evaluating

$$\mathbb{P}(-1 + 1 - p + \xi \leq 0) = \mathbb{P}(\xi \leq p) = p \geq p.$$

By the definition of X, the optimal solution of the chance constrained optimization problem therefore has to be $x^* = -1$.

To approximate the problem let us define a sequence of inner approximation problems by $\Delta_k := \{0\} \subseteq \mathbb{R}$ and $D_k : X \times \Delta_k \to \mathcal{A}, D_k(x, \delta) = \left[0, p + \frac{1}{k}\right]$, such that Δ_k is compact for all $k \in \mathbb{N}$ and D_k is continuous as a constant mapping for all $k \in \mathbb{N}$.

Fixing $x_0 := 1$ *and an arbitrary* $k \in \mathbb{N}$*, we check*

$$\begin{aligned}\mathbb{P}(D(x_0,\delta)) &= \mathbb{P}\left(\left[0, p+\frac{1}{k}\right]\right)\\ &= \min\left\{p+\frac{1}{k}, 1\right\} \geq p\end{aligned}$$

and

$$\begin{aligned}\max_{\xi \in D(x_0,\delta)} g(x_0,\xi) &= \max_{\xi \in \left[0, p+\frac{1}{k}\right]} -1-1-p+\xi\\ &= -2-p+p+\frac{1}{k}\\ &= -2+\frac{1}{k} \leq 0.\end{aligned}$$

Consequently, $x = 1$ *is feasible for all* $k \in \mathbb{N}$ *independent from* $p \in [0,1]$ *and therefore* $p \leq p_{\max}(\mathcal{D}_k) = 1$*, where* $\mathcal{D}_k := \bigcup_{x \in X, \delta \in \Delta_k} D_k(x,\delta) = D_k$*.*
Additionally, $\Omega(x^*) = \{\xi \in \Xi \mid g(x^*,\xi) \leq 0\} = \{\xi \in [0,1] \mid -p+\xi \leq 0\} = [0,p]$ *is known and we can guarantee*

$$\begin{aligned}\lim_{k\to\infty} d_H(D_k,\Omega(x^*)) &= \lim_{k\to\infty} d_H\left(\left[0,p+\frac{1}{k}\right],[0,p]\right) = 0 \text{ and}\\ \lim_{k\to\infty} d_{\mathbb{P}}(D_k,\Omega(x^*)) &= \lim_{k\to\infty} d_{\mathbb{P}}\left(\left[0,p+\frac{1}{k}\right],[0,p]\right) = 0.\end{aligned}$$

Unfortunately, the feasible set of decision for any fixed $k \in \mathbb{N}$ *is*

$$\tilde{\mathcal{F}}_{D_k,p} = \left[\frac{\sqrt{1+\frac{4}{k}}-1}{2}, 1\right] \subsetneq [0,1].$$

This means that there cannot be any sequence $(y_k)_{k\in\mathbb{N}} \subseteq X$ *such that* $y_k \in \tilde{\mathcal{F}}_{D_k,p}$ *and* $\lim_{k\to\infty} y_k = x^* = -1$*.*

To make such a sequence of set-approximation problems converge towards a solution of the corresponding chance constrained optimization problem is the focus of the next section.

3.3 Set-approximation algorithm for probust optimization problems

In the last section, we have seen that we can find feasible decisions for a chance constrained optimization problem by solving its corresponding inner set-approximation for a given design space Δ and design function D.

In this section, we describe an iterative algorithm for increasing design spaces and study its convergence behavior as well as its modification to be used on probust optimization problems.

Before noting the algorithm, we need a sequence of design spaces $(\Delta_k)_{k\in\mathbb{N}}$ and a sequence of designs $(D_k)_{k\in\mathbb{N}}$ to describe it: In analogy to probust subset schemes, we analyze the simpler case, where we have a monotone behavior with respect to the design spaces.

Definition 3.3.1 (Increasing set-approximation schemes)
We call a sequence of design spaces and design functions $\Psi = (\Delta_k, D_k)_{k\in\mathbb{N}}$ a set-approximation scheme.
If a set-approximation scheme Ψ satisfies

$$\forall x \in X, k \in \mathbb{N}, \delta_k \in \Delta_k \exists \delta_{k+1} \in \Delta_{k+1} : D_k(x, \delta_k) = D_{k+1}(x, \delta_{k+1}),$$

we call the set-approximation scheme Ψ an increasing set-approximation scheme.
We can represent the family of designs corresponding to an increasing set-approximation scheme Ψ by

$$\mathcal{D}_\Psi = \bigcup_{k\in\mathbb{N}} \mathcal{D}_k = \bigcup_{k\in\mathbb{N}} \left(\bigcup_{x\in X, \delta\in\Delta_k} D_k(x, \delta) \right).$$

With these definitions, we can describe the iterative set-approximation algorithm as:

Algorithm 4 Set-approximation algorithm for chance constrained optimization problems

```
Inputs:
    Chance constrained optimization problem instance $CC$,
    set-approximation scheme $\Psi$, initial decision $x_0 \in X$
Initialize:
    $k := 0$
do
    $(x_{k+1}, \delta_{k+1}) \leftarrow \arg\min_{(x,\delta)\in X\times\Delta_k} f(x)$ s.t. $\mathbb{P}(D_k(x,\delta)) \geq p$,
                                                          $\max_{\xi\in D_k(x,\delta)} g(x,\xi) \leq 0$
    $k \leftarrow k + 1$
while stopping criterion is not fulfilled
Results:
    Sequence $(x_k, \delta_k)_{k\in\mathbb{N}}$
```

In comparison to the probust subset algorithm 3, we have no update of the design space or design function influenced by the current decision x_k or the current design parameter δ_k such that we are considering pre-defined set-approximations instead of adaptive ones. The problem here is on the modelling side as a function that maps the current iterates (x_k, δ_k) to a pair of a design space with corresponding design function is quite hard to handle. We will get back to this point defining sandwiching algorithms in the next chapter.

With the defined set-approximation algorithm 4, we want to focus on its convergence behavior.

We have already seen in Example 3.2.3 that we cannot expect convergence in general.

The drawback of inner approximations is that we need to guarantee that we can find decisions that are on the one hand close to the optimal solution x^* and on the other hand feasible for some iterations $k \in \mathbb{N}$. A condition to ensure that in nonlinear optimization problems is, i.e., the Mangasarian-Fromovitz constraint qualification (MFCQ), see Lemma 1.2.10.

The next lemma shows that if an optimal solution of the chance constrained optimization problem fulfills the MFCQ, then we can guarantee the convergence of an iterative inner set-approximation if the design spaces are chosen wisely.

Nevertheless, we point out that MFCQ has nice implications when it holds, despite it being hard to verify because this condition is based on gradients which are rarely given for probability evaluation functions.

Theorem 3.3.2 (Convergence theorem of inner set-approximation for CC) *Let X be a compact set, $f : X \to \mathbb{R}$ be a continuous function, $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $g : X \times \Xi \to \mathbb{R}$ be a Caratheodory-function, $p \in [0, 1]$, $(\Delta_k)_{k\in\mathbb{N}}$ be a sequence of compact sets, $D_k : X \times \Delta_k \to \mathcal{A}$ design functions such that*

- *$\Gamma_k : X \times \Delta_k \rightrightarrows \Xi, (x, \delta) \mapsto D_k(x, \delta)$ are continuous correspondences for all $k \in \mathbb{N}$ and*
- *$D_k(x, \Delta_k) \subseteq D_{k+1}(x, \Delta_{k+1})$ for all $x \in X$, meaning that for each $x \in X$ and each $\delta_k \in \Delta_k$, we can find a $\delta_{k+1} \in \Delta_{k+1}$ such that $D_k(x, \delta_k) = D_{k+1}(x, \delta_{k+1})$.*

Furthermore, let $x^* \in X$ be a minimizer of the chance constrained optimization problem. If there exists a MFCQ-vector for the original chance constrained optimization problem at x^* and a radius $r > 0$ such that for all $x \in B_r(x^*) \cap \mathcal{F}_p$ there exists a sequence $(\delta_k(x))_{k\in\mathbb{N}}, \delta_k : X \to \Delta_k$ for all $k \in \mathbb{N}$ such that

- *$\lim_{k\to\infty} d_{\mathbb{P}}(\Omega(x), D_k(x, \delta_k(x))) = 0$ and*
- *$D_k(x, \delta_k(x)) \subseteq \Omega(x)$ for all $k \in \mathbb{N}$,*

then every accumulation point of the set-approximation algorithm is a minmizer of the original chance constrained optimization problem.

Proof. We will show this claim by constructing a sequence of points $(\hat{x}_n)_{n\in\mathbb{N}}$ that converges towards x^* and for each $n \in \mathbb{N}$, we can find a $k_n \in \mathbb{N}$ with $k_n \geq n$ and $\hat{x}_n \in \tilde{\mathcal{F}}_{D_{k_n},p}$. Because we assumed that $D_k(\hat{x}_n, \Delta_k) \subseteq D_{k+1}(\hat{x}_n, \Delta_{k+1})$ for all $n \in N, k \in \mathbb{N}$, we know that $\hat{x}_n \in \tilde{\mathcal{F}}_{D_k,p}$ for all $k \geq k_n$ and there exists a smallest $k_n \in \mathbb{N}$ satisfying $\hat{x}_n \in \tilde{\mathcal{F}}_{D_{k_n},p}$. Consequently, we can estimate

$$f(x^*) = \min_{x\in\mathcal{F}_p} f(x) \leq \min_{x\in\tilde{\mathcal{F}}_{D_k,p}} f(x) = f(x_k) \leq f(\hat{x}_n) \ \forall n \in \mathbb{N}, k \geq k_n.$$

Because $f(\hat{x}_n)$ converges towards $f(x^*)$ for $n \to \infty$, the sequence $(f(x_k))_{k\in\mathbb{N}}$ has also to converge towards $f(x^*)$ for $k \to \infty$. As this value is the (global) minimum, we know that all accumulation points $\overline{x}$ of $(x_k)_{k\in\mathbb{N}}$ are (global) minima.
Now we construct the sequence of points $(\hat{x}_n)_{n\in\mathbb{N}}$:
Since we assumed that φ is twice continuously differentiable in a neighborhood of x^*, there exists a radius $R > 0$ such that $B_R(x^*)$ lies within this neighborhood. We denote

$$c := \max_{h\in[-R,R]} |\langle v, D^2\mathbb{P}(\Omega(x^* + hv))v\rangle|,$$

where D^2 denotes the Hessian of the probability evaluation function. Introducing the variable $H := \min\{r, R, \frac{1}{c}\}$ and choosing any $h \in [0, H]$, we can estimate the probability evaluation by a Taylor-expansion with Lagrange remainder and the definition of the MFCQ-vector via

$$\begin{aligned}\mathbb{P}(\Omega(x^* + hv)) &= \mathbb{P}(\Omega(x^*)) + \nabla_x\mathbb{P}(\Omega(x^*))hv + \frac{\langle v, D^2\mathbb{P}(\Omega(x^* + \tilde{h}v))v\rangle}{2}h^2 \\ &\geq p + h - \frac{ch^2}{2} \geq p + \frac{h}{2},\end{aligned}$$

where $\tilde{h} \in [0, h] \subseteq [0, R]$. Consequently, we know that $x^* + hv \in \mathcal{F}_p$ for all $h \in [0, H]$. Because we assumed that there exists a $r > 0$ such that for all $x \in B_r(x^*) \cap \mathcal{F}_p$ we can approximate $\Omega(x)$ by $(D_k(x, \delta_k(x))_{k\in\mathbb{N}}$ (w.r.t. $d_\mathbb{P}$). For $0 < h \leq H \leq r$ and $x = x^* + hv$ we can therefore estimate

$$\begin{aligned}\mathbb{P}(D_k(x^* + hv, \delta_k(x^* + hv))) &\geq \mathbb{P}(\Omega(x^* + hv)) - \epsilon_k \\ &\geq p + \frac{h}{2} - \epsilon_k.\end{aligned}$$

Because $D_k(x^* + hv, \Delta_k) \subseteq D_{k+1}(x^* + hv, \Delta_{k+1})$ for all $k \in \mathbb{N}$, we know that $(\epsilon_k)_{k\in\mathbb{N}}$ is a monotonically decreasing sequence with limit 0. Consequently, we can find a minimal index $K(h) \in \mathbb{N}$ such that $k \geq K(h)$ implies $\epsilon_k \leq \frac{h}{2}$ and therefore

$$\mathbb{P}(D_k(x^* + hv, \delta_k(x^* + hv))) \geq p.$$

As we assumed that $D_k(x, \delta_k(x)) \subseteq \Omega(x)$ holds for all $x \in B_r(x^*) \cap \mathcal{F}, k \in \mathbb{N}$, we can verify that

$$\max_{\xi\in D_k(x^*+hv,\delta_k(x^*+hv))} g(x^* + hv, \xi) \leq \max_{\xi\in\Omega(x^*+hv)} g(x^* + hv, \xi) \leq 0.$$

Consequently, the step size $h_n := \frac{H}{n}$ leads to an iteration $k_n := \max\{K(h_n), k_{n-1} + 1\}$ such that $\hat{x}_n := x^* + \frac{H}{n} \in \tilde{\mathcal{F}}_{D_{k_n},p}$ for all $n \in \mathbb{N}$. Using the first part of the proof the claim holds. □

The assumptions of this convergence result imply that there is a direction v such that we can approximate $\Omega(x^*)$ from this direction with sets $(D_k(\hat{x}_k, \hat{\delta}_k))$, where $(\hat{x}_k, \hat{\delta}_k) \in \mathcal{F}_{D_k,p}$ are feasible decision w.r.t. the kth approximation problem.

Please note that we can only guarantee to achieve an objective value of decisions $x \in X$ whose set of feasible realizations can be approximated by $(D_k(x, \delta))_{k\in\mathbb{N},\delta\in\Delta_k}$ sufficiently well. Since in general we do not know which decision leads to the global minimum, we have to approximate the set of feasible realizations for as much decisions as possible. How these sets are structured is already mentioned in Proposition 3.1.7 for fixed decisions $x \in X$.

After we have understood how the set-approximation algorithm works for chance constrained optimization problems, we now are interested in using it for probust optimization problems. To extend the concept, we just have to use Berge's maximum Theorem 1.2.8. Since the inner function of a probust optimization problem is continuous w.r.t. $t \in T$ for all $(x, \xi) \in X \times \Xi$ and the set T is compact, we know that the function $g(x, \xi, T) = \max_{t\in T} g(x, \xi, t)$ is continuous. Consequently, we can interpret probust optimization problems as a special instances of chance constrained optimization problems and use all results from the chance constrained case.
The adapted version of the set-approximation algorithm for probust optimization problems looks like

Algorithm 5 Set-approximation algorithm for probust optimization problems

```
Inputs:
    (Standard) probust optimization problem instance SPP,
    set-approximation scheme $\Psi$, initial decision $x_0 \in X$
Initialize:
    $k := 0$
do
    $(x_{k+1}, \delta_{k+1}) \leftarrow \arg\min_{(x,\delta)\in X\times\Delta_k} f(x)$ s.t. $\mathbb{P}(D_k(x, \delta)) \geq p$,
                                                            $\max_{\xi\in D_k(x,\delta),t\in T} g(x, \xi, t) \leq 0$
    $k \leftarrow k + 1$
while stopping criterion is not fulfilled
Results:
    Sequence $(x_k, \delta_k)_{k\in\mathbb{N}}$
```

We state the convergence of the inner set-approximation approach for standard probust optimization problems as another theorem.

Theorem 3.3.3 (Convergence theorem of inner set-approximation for SPP)
Let X, T be compact sets, $f : X \to \mathbb{R}$ be a continuous function, $(\Xi, \mathcal{A}, \mathbb{P})$ be a probability space, $g : X \times \Xi \times T \to \mathbb{R}$ be an inner function, $p \in [0, 1]$, $(\Delta_k)_{k\in\mathbb{N}}$ be a sequence of compact sets, $D_k : X \times \Delta_k \to \mathcal{A}$ design functions such that

- *$\Gamma_k : X \times \Delta \rightrightarrows \Xi, (x, \delta) \mapsto D_k(x, \delta)$ are continuous correspondences for all $k \in \mathbb{N}$*
- *$D_k(x, \Delta_k) \subseteq D_{k+1}(x, \Delta_{k+1})$ for all $x \in X$, meaning that for each $x \in X$ and each $\delta_k \in \Delta_k$, we can find a $\delta_{k+1} \in \Delta_{k+1}$ such that $D_k(x, \delta_k) = D_{k+1}(x, \delta_{k+1})$.*

Furthermore, let $x^* \in X$ be a minimizer of the chance constrained optimization problem. If there exists a MFCQ-vector for the original chance constrained optimization problem at x^* and a radius $r > 0$ such that for all $x \in B_r(x^*) \cap \mathcal{F}_p$ there exists a sequence $(\delta_k(x))_{k\in\mathbb{N}}, \delta_k \in \Delta_k$ for all $k \in \mathbb{N}$ such that

- $\lim_{k\to\infty} d_{\mathbb{P}}(\Omega(x), D_k(x, \delta_k(x))) = 0$ *and*

- $D_k(x, \delta_k(x)) \subseteq \Omega(x)$ *for all* $k \in \mathbb{N}$*,*

then every accumulation point of the set-approximation algorithm is a minimizer of the original standard probust optimization problem.

Proof. We will show this claim using Theorem 3.3.2.
As we assumed that g is continuous w.r.t. $(x, \xi, t) \in X \times \Xi \times T$ and T is compact, we know by Berge's maximum Theorem 1.2.8 that

$$g(x, \xi, T) := \max_{t\in T} g(x, \xi, t)$$

is a continuous function w.r.t. $(x, \xi) \in X \times \Xi$.
Consequently, $g(x, \cdot, T)$ is Borel-measurable for all $x \in X$ and $g(\cdot, \xi, T)$ is continuous for all $\xi \in \Xi$ and therefore a Caratheodory-function.
Now we can use Theorem 3.3.2 to determine a solution of the chance constrained optimization problem

$$\min_{x\in X} f(x) \text{ s.t. } \mathbb{P}(g(x, \xi, T) \leq 0) \geq p.$$

Because we just reformulated the inequality constraint, the minimizer of this chance constrained optimization problem is the minimizer of the original standard probust optimization problem and the claim holds. □

After introducing an iterative set-approximation scheme to solve standard probust optimization problems in this section, we concretize this concept by applying one iteration of the process to an example problem in the next section.

3.4 Example of how to use a set-approximation scheme

In this section, we define a special probust optimization problem instance, we formulate the corresponding inner set-approximation problem by fixing a design function and solve this inner set-approximation problem. This problem instance is a stochastic design-centering problem which we will introduce later in Chapter 5.

Problem definition

We consider the problem of putting an area maximizing circle

$$B_r(M) = \{y \in \mathbb{R}^2 \mid (y_1 - m_1)^2 + (y_2 - m_2)^2 \leq r^2\}$$

with $r \geq 0, M \in \mathbb{R}^2$ into an uncertain triangle that is described by the points

$$\begin{aligned} P_1 &= (Z_1, 0), Z_1 \sim \mathcal{U}\left([1,2]\right), \\ P_2 &= (-Z_2, 0), Z_2 \sim \mathcal{U}\left([1,3]\right), \\ P_3 &= (0, Z_3), Z_3 \sim \mathcal{N}\left(2,1\right). \end{aligned}$$

Note that this triangle is degenerated if the realization of Z_3 is zero. Given the distribution of this random variable, this happens with probability 0 though.
We can describe the triangle for a fixed realization $\xi \in \Xi := \mathbb{R}^3$ as

$$\begin{aligned} C(\xi) = \{y \in \mathbb{R}^2 \mid g_1(\xi, y) &\leq 0, g_2(\xi, y) \leq 0, g_3(\xi, y) \leq 0\}, \\ \text{where } g_1(\xi, y) &= -\text{sign}(\xi_3) y_2, \\ g_2(\xi, y) &= \text{sign}(\xi_3)\left(\xi_1 y_2 + \xi_3 y_1 - \xi_1 \xi_3\right), \\ g_3(\xi, y) &= \text{sign}(\xi_3)\left(\xi_2 y_2 - \xi_3 y_1 - \xi_2 \xi_3\right). \end{aligned}$$

With fixed $p := 0.9$ the stochastic design-centering problem can be written as

$$\text{SDC}: \max_{r \geq 0, M \in \mathbb{R}^2} \pi r^2 \text{ s.t. } \mathbb{P}(B_r(M) \subseteq C(\xi)) \geq 0.9.$$

Problem reformulation as a standard probust optimization problem

Before solving this problem, we have to reformulate it as a standard probust optimization problem to be able to define and solve the inner set-approximation problem introduced in this chapter.
In a first step, we rewrite SDC as a generalized probust optimization problem. How we can do that is explained in the beginning of Chapter 5:

$$\text{SDC-GPP}: \min_{r \geq 0, M \in \mathbb{R}^2} -\pi r^2 \text{ s.t. } \mathbb{P}\begin{pmatrix} g_1(\xi, y) \leq 0 \ \forall y \in B_r(M), \\ g_2(\xi, y) \leq 0 \ \forall y \in B_r(M), \\ g_3(\xi, y) \leq 0 \ \forall y \in B_r(M) \end{pmatrix} \geq 0.9$$

Next, we define the set of decisions $X := \mathbb{R}_{\geq 0} \times \mathbb{R}^2$ and use the following transformation to the reference set $\hat{T} := B_1(0)$ to simplify the set-dependency within the probability evaluation

$$\mathcal{T}_{\hat{T}} : X \times \Xi \times \hat{T} \to \mathbb{R}^2, ((r, M), \xi, z) \to rz + M.$$

As $\mathcal{T}_{\hat{T}}(x,\xi\cdot)$ is surjective for all $(x,\xi) \in X \times \Xi$ and even a homeomorphism between $\hat{T} = B_1(0)$ and $B_r(M)$ if $r \neq 0$, we can reformulate the inner functions of SDC-GPP as

$$\begin{aligned}
\tilde{g}_1(x,\xi,z) &= g_1(\xi,\mathcal{T}_{\hat{T}}(x,\xi,z)) = -(rz_2+m_2),\\
\tilde{g}_2(x,\xi,z) &= g_2(\xi,\mathcal{T}_{\hat{T}}(x,\xi,z)) = \xi_1(rz_2+m_2)+\xi_3(rz_1+m_1)-\xi_1\xi_3,\\
\tilde{g}_3(x,\xi,z) &= g_3(\xi,\mathcal{T}_{\hat{T}}(x,\xi,z)) = \xi_2(rz_2+m_2)-\xi_3(rz_1+m_1)-\xi_2\xi_3,
\end{aligned}$$

while keeping the feasible decisions and minimizers the same according to Theorem 1.3.13. This leads to the following standard probust optimization problem

$$\text{SDC-SPP}: \min_{r\geq 0, M\in\mathbb{R}^2} -\pi r^2 \text{ s.t. } \mathbb{P}\begin{pmatrix}\tilde{g}_1(x,\xi,z)\leq 0\ \forall z\in B_1(0),\\ \tilde{g}_2(x,\xi,z)\leq 0\ \forall z\in B_1(0),\\ \tilde{g}_3(x,\xi,z)\leq 0\ \forall z\in B_1(0)\end{pmatrix} \geq 0.9.$$

Please note that we ignored the sign-functions in the definition of $\tilde{g}_i, i=1,2,3$.
We are allowed to do so because any feasible decision $x=(r,M)$ with $r>0$ satisfies $\xi_3\geq 0$ for all its feasible realizations $\xi\in\Omega(x)$ as we see by the following indirect argument that leads to a case distinction:
Given $x\in X, r>0$ let us assume there exists a feasible realization $\xi\in\Omega(x)$ with $\xi_3<0$.
1. Case: There exists another feasible realization $\tilde{\xi}\in\Omega(x)$ with $\tilde{\xi}_3\geq 0$ which implies $B_r(M)\cap C(\xi)=\emptyset$ or $B_r(M)\cap C(\tilde{\xi})=\emptyset$ contradicting $\xi,\tilde{\xi}\in\Omega(x)$. A sketch of this argument is shown in Figure 3.1.
2. Case: All feasible realizations satisfy $\xi_3<0$ and we can therefore estimate

$$\varphi(x)\leq\mathbb{P}(\xi_3\leq 0)=F(-2)\approx 0.0228<p=0.9.$$

In both cases the decision $x\in X$ cannot be feasible and consequently all feasible decisions imply $\xi_3\geq 0$ for all $\xi\in\Omega(x)$.

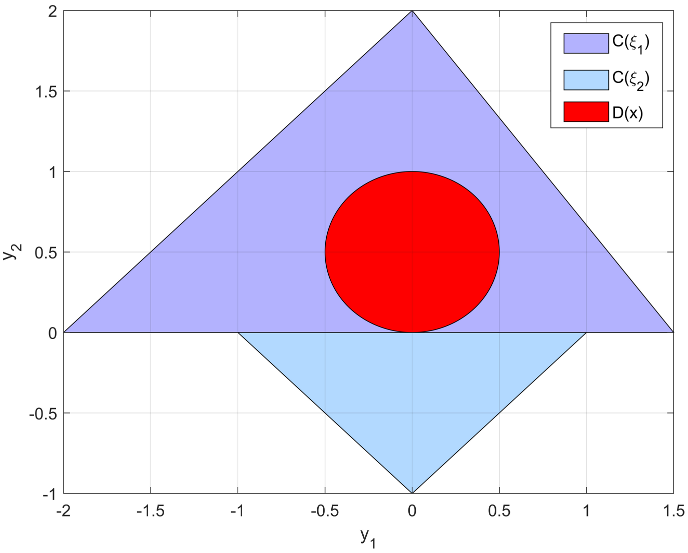


Figure 3.1: Visualization of two triangle realizations $C(\xi_1), C(\xi_2)$ with positive and negative height and some circle $D(x)$

Choosing an appropriate set-approximation scheme

After rewriting the stochastic design-centering problem as a standard probust optimization problem, we have to think about the structure of the set of feasible realizations to define an appropriate design function for the inner set-approximation problem.
We show that the inner functions $\max_{z\in B_1(0)} \tilde{g}_i(x,\xi,z)$ are monotonically increasing w.r.t. to the components of ξ if we fix a feasible $x \in X$.
Therefore, we fix a feasible decision $x \in \mathcal{F}_p$, a feasible realization $\xi \in \Omega(x)$ and show that $\tilde{g}(x,\xi,z) \geq \tilde{g}(x,\tilde{\xi},z)$ for all $z \in B_1(0)$ and $\tilde{\xi} \geq \xi$ (component-wise).
First we consider $\tilde{g}_1(x,\xi,z) = -(rz_2+m_2)$: Because this constraint does not depend on any ξ-component, the monotonicity assumption holds for $\tilde{g}_1$.
In the next step, we consider the inner function $\tilde{g}_2(x,\cdot,z)$. Because $\xi \in \Omega(x)$, we can check the monotonicity componentwise:

$$\begin{aligned}
j=1:\ \tilde{g}_2(x,\xi,z) &= \xi_1(\underbrace{rz_2+m_2}_{\leq \xi_3 \text{ because } B_r(M)\subseteq C(\xi)}) + \xi_3(rz_1+m_1) - \xi_1\xi_3 \\
&= \underbrace{\xi_1}_{\geq 0}\underbrace{(rz_2+m_2-\xi_3)}_{\leq 0} + \xi_3(rz_1+m_1) \\
&\geq \tilde{\xi}_1(rz_2+m_2-\xi_3) + \xi_3(rz_1+m_1) \\
&= \tilde{g}_2(x,(\tilde{\xi}_1,\xi_2,\xi_3),z)\ \forall \tilde{\xi}_1 \geq \xi_1 \\
j=2:\ \tilde{g}_2(x,\xi,z) &= \xi_1(rz_2+m_2) + \xi_3(rz_1+m_1) - \xi_1\xi_3 \\
&= \tilde{g}_2(x,(\xi_1,\tilde{\xi}_2,\xi_3),z)\ \forall \tilde{\xi}_2 \geq \xi_2 \\
j=3:\ \tilde{g}_2(x,\xi,z) &= \xi_1(rz_2+m_2) + \xi_3(\underbrace{rz_1+m_1}_{\leq \xi_1 \text{ because } B_r(M)\subseteq C(\xi)}) - \xi_1\xi_3 \\
&= \xi_1(rz_2+m_2) + \xi_3\underbrace{(rz_1+m_1-\xi_1)}_{\leq 0} \\
&\geq \xi_1(rz_2+m_2) + \tilde{\xi}_3(rz_1+m_1-\xi_1) \\
&= \tilde{g}_2(x,(\xi_1,\xi_2,\tilde{\xi}_3),z)\ \forall \tilde{\xi}_3 \geq \xi_3
\end{aligned}$$

Since we can decompose $\tilde{\xi} \geq \xi$ in component-wise inequalities, the claim holds for $\tilde{g}_2$ for all $\tilde{\xi} \geq \xi$. Analogously, we can show the same result for $\tilde{g}_3$.
Because all components $\tilde{g}_i, i = 1,2,3$ are monotonically increasing w.r.t. ξ, this is also true for $\tilde{g}$ as the maximum over these functions. As the $z \in Z$ was arbitrarily fixed, we know that the function $\max_{z\in B_1(0)} \tilde{g}(x,\cdot,z)$ is also monotonically increasing w.r.t. ξ.
Consequently, using Proposition 3.1.7 (iv) the set of feasible realizations can be described by intervals $[\underline{\xi}_i,\infty) \cap \Xi_i, \underline{\xi}_i \in \mathbb{R}$ concerning one fixed entry $i = 1,2,3$ of $\xi \in \mathbb{R}^3$, if the other two entries are fixed. Consequently, a promising approach is

$$\begin{aligned}
&\Delta := \{\delta \in \mathbb{R}^3\ |\delta_1 \in [1,2], \delta_2 \in [1,3], \delta_3 \in [0,\infty)\}, \\
&D(x,\delta) := [\delta_1,2] \times [\delta_2,3] \times [\delta_3,\infty).
\end{aligned}$$

Be aware that the design function is independent of $x \in X$.

Because the components of the random vector Z were assumed to be uncorrelated, we can evaluate the probability $\mathbb{P}(D(x,\delta))$ by evaluating the probabilities of the single components:

$$\begin{aligned}\mathbb{P}(D(x,\delta)) &= \mathbb{P}_1(\xi \geq \delta_1)\cdot \mathbb{P}(\xi_2 \geq \delta_2)\cdot \mathbb{P}_3(\xi \geq \delta_3)\\ &= (2-\delta_1)\cdot\left(\frac{3-\delta_2}{2}\right)\cdot(1-F(\delta_3-2))\end{aligned}$$

This means we can reformulate the probability evaluation constraint of the inner set-approximation problem as an analytical function and the remaining problem is a semi-infinite optimization problem that can be solved, e.g., by the adaptive discretization of Blankenship and Falk [17].
The solution of this problem is

$$\begin{aligned}&f(x^*) \approx -0.3257,\\ &x^* \approx (0.322, 0, 0.322)\\ &\delta^* = (1,1,2-F^{-1}(0.1)) \approx (1,1,0.7184).\end{aligned}$$

By Proposition 3.2.2 and Theorem 1.3.13 we know that this solution is also a feasible decision of the stochastic design-centering SDC. The solution x^* is not (exactly) the optimal solution of the original probust optimization problem because there exist feasible realizations $\xi \in \Omega(x^*)$ that fulfill, e.g., $\xi_1, \xi_2 > 1$, $\xi_3 < 2 - F^{-1}(0.1)$ which are not considered by the design D.
A visualization of the optimal circle in a reference triangle that corresponds to the value δ^* can be seen in the Figure 3.2.

How we can solve probust optimization problems where we cannot find the set of feasible realizations directly and instead have to use iterative set-approximation algorithm is the focus of the next chapter. We will combine our results from probust subset schemes and set-approximation schemes into what we will call sandwiching algorithms.

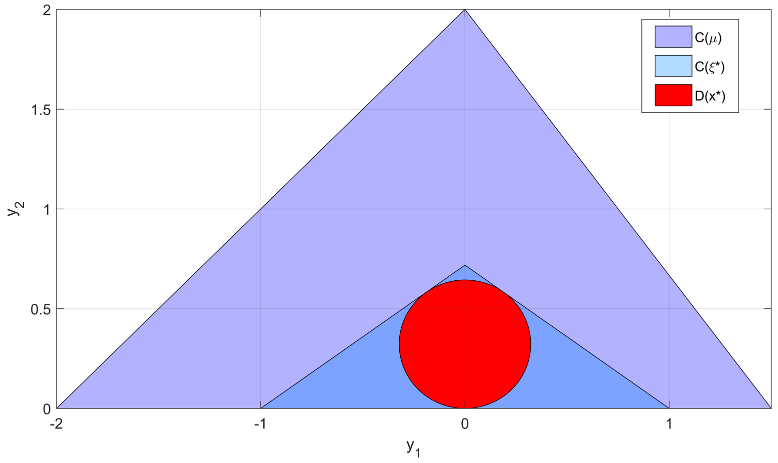


Figure 3.2: Visualization of expected triangle $C(\mu)$, reference triangle $C(\xi^*)$, where $\xi^* = (1,1,2-F^{-1}(0.1))$ and solution circle $D(x^*)$

4 Sandwiching algorithms

In the last two chapters, we have introduced concepts to solve standard probust optimization problems approximately. While the probust subset algorithm 3 calculates lower bounds for the minimal objective value, the set-approximation algorithm 5 calculates upper bounds.

In this chapter, we present how we can combine these two algorithms to use their individual strengths and cover their weaknesses.

We start with the following generic sandwiching algorithm:

Algorithm 6 Sandwiching algorithm for standard probust optimization problems

```
Inputs:
    Probust optimization problem instance SPP, precision $\epsilon > 0$,
    discretization scheme $\Phi$, set-approximation scheme $\Psi$,
    initial decision $x_0 \in X$, initial discretization $T_0 \subseteq T$
Initialize:
    $k := 0$
do
    $\underline{x}_{k+1} \leftarrow \arg\min_{x \in X} f(x)$ s.t. $\varphi(x, T_k) \geq p$
    $T_{k+1} \leftarrow \Phi_k(\underline{x}_{k+1}, T_k)$
    $(\overline{x}_{k+1}, \delta_{k+1}) \leftarrow \arg\min_{(x,\delta) \in X \times \Delta_{k+1}} f(x)$ s.t. $\mathbb{P}(D_{k+1}(x, \delta)) \geq p$,
                                                  $\max_{\xi \in D_{k+1}(x,\delta), t \in T} g(x, \xi, t) \leq 0$
    $k \leftarrow k + 1$
while $f(\overline{x}_k) - f(\underline{x}_k) \geq \epsilon$
Results:
    Sequence $(\underline{x}_k, \overline{x}_k, T_k, \delta_k)_{k \in \mathbb{N}}$
```

We call this algorithm generic, because we use both iterative Algorithms 3 and 5 without specifying the probust subset schemes or the set-approximation scheme. Since both solution schemes do not exchange calculated values such as $\underline{x}_k, \overline{x}_k, T_k$ or δ_k for any $k \in \mathbb{N}$, they define bounds for the optimal objective value by calculating $f(\underline{x}_k)$ and $f(\overline{x}_k)$ like working on their own. The only advantage we gain in comparison to the individual algorithms is that calculating these bounds allows us to define a stopping criterion by $f(\overline{x}_k) - f(\underline{x}_k) < \epsilon$ for a given precision $\epsilon > 0$.

Because the solution schemes in this algorithm are separated from each other, we can add the assumptions for the convergence results in Theorem 2.2.3 and Theorem 3.3.2 to directly show the convergence of Algorithm 6 after finitely many steps.

How we can exchange information between both solution schemes to benefit from their strong sides while covering their weaknesses is the main question of the next sections.

4.1 Probust subset schemes using set-approximation information

In this section, we focus on using information generated by probust subset schemes to define set-approximation schemes. We have seen in Chapter 3 that set-approximation schemes do find decisions that are feasible w.r.t. to the corresponding probust optimization problem. The drawback of these methods is that we cannot guarantee to find the optimal solution of the corresponding problem and we have to predefine the family of designs $\mathcal{D}$. Furthermore, we have not specified a stopping criterion in Algorithm 3 or 5 so far.

The main idea of this section is to use the generated iterates $(\underline{x}_k)_{k\in\mathbb{N}}$ from probust subset schemes to help us define a stopping criterion, while we use the sequence $(T_k)_{k\in\mathbb{N}}$ to define designs for the set-approximation scheme. This way the design function D_k adapts in each iteration $k \in \mathbb{N}$ to the current information $\underline{x}_k, T_k$. As mentioned in the last chapter, this is not given in general for set-approximation schemes.

In the following, we assume we are using an increasing subset scheme Φ that satisfies the assumptions of Theorem 2.2.3. Since the theorem guarantees that we find all "interesting" scenarios $S = \lim_{k\to\infty} T_k \subseteq T$ for the accumulation points $\bar{x} \in X$ of the defined sequence $(\underline{x}_k)_{k\in\mathbb{N}}$, we hope that the corresponding inner set-approximations does also converge.

To get a feeling for the convergence behavior of such methods, we test the following designs in the next example:

1. We use designs specified by

$$D_k(x, \delta) := \Omega(y, T_k) \tag{4.1}$$

for some $y \in X$. This way D_k is independent of x and we can identify the design parameter δ by the tuple (y, T_k) for some $y \in X$. We do not use $D_k(x, \delta) = \Omega(x, T_k)$ because for $\Omega(x, T_k) \subsetneq \Omega(x, T) = \Omega(x)$, we cannot find a feasible decision for the constraint (3.4) in general.

2. We use the design specified by

$$D_k(x, \delta) := \Omega(\underline{x}_k, T_k), \tag{4.2}$$

where $\underline{x}_k$ is the solution of the kth iteration of the probust subset scheme. Here again, we make the design D_k independent of x and do also fix the design parameter $\delta = (\underline{x}_k, T_k)$ in the single iterations of the set-approximation algorithm. This has two advantages: First, we know by definition of $\underline{x}_k$ that $\mathbb{P}(\Omega(\underline{x}_k, T_k)) \geq p$. Thus, we can ignore the probability evaluation constraint (3.3) in the inner set-approximation. Second, the maximum constraint (3.4) simplifies from a generalized semi-infinite constraint to a standard semi-infinite constraint. This constraint can be handled, e.g., by the adaptive algorithm introduced by Seidel in [73].

3. We use designs specified by

$$D_k(x,\delta) \coloneqq \Omega_r(x,T_k) \coloneqq \{\xi \in \Xi \mid g(x,\xi,t) + r \leq 0 \ \forall t \in T_k\}, \tag{4.3}$$

   where $r \in \mathbb{R}$ together with T_k can be identified with the design parameter δ.
   We can interpret $r \in \mathbb{R}$ as a balancing factor between the constraints in the inner set-approximation problem:
   For $r > 0$ we know by definition $\Omega_r(x,T_k) \subseteq \Omega(x,T_k)$. Consequently, it is harder to fulfill the probability condition (3.3) in the inner set-approximation problem for such sets. Nevertheless, the relation $\Omega_r(x,T_k) \subseteq \Omega(x,T_k)$ does also imply that the maximum condition (3.4) of the inner set-approximation is easier to satisfy. It is the other way around if we choose $r < 0$.

A lot of other design definitions are also possible. Going through all of them is not within the range of this thesis.

Within the range of this thesis is testing the convergence of the solutions of set-approximation problems using the defined designs in the following example:

Example 4.1.1 (Example: Convergence of set-approximation schemes)
We consider the following probust optimization problem

$$\min_{x\in[0,1]} x \ \textit{s.t.}\ \mathbb{P}(x-\xi \leq 0, \xi - 2x - 1.9 + t \leq 0 \ \forall t \in [0,1]) \geq 0.9, \ \textit{where } Z \sim \mathcal{U}\left([0,1]\right).$$

First, we solve this problem analytically like in Example 1.1.9. We know that the worst-case function evaluation is defined by

$$\max_{t\in[0,1]} \xi - 2x - 1.9 + t = \xi - 2x - 0.9$$

for arbitrary $(x,\xi) \in [0,1]^2$. Therefore, we know that the unique worst-case scenario is $t = 1$.
This leads to the description of the set of feasible realizations

$$\Omega(x) = \Omega(x,T) = \Omega(x,\{1\}) = [x, \min\{2x+0.9, 1\}] \subseteq \Xi = [0,1]$$

for any fixed $x \in [0,1]$. As the uniform distribution can be evaluated easily in this case, we can reformulate the probust constraint as for $x \in [0,1]$ as

$$\varphi(x) = \mathbb{P}(\Omega(x)) = \min\{2x+0.9, 1\} - x \geq 0.9.$$

This leads to the (unique) minimizer $x^* = 0$ with $\varphi(x^*) = \min\{0.9, 1\} - 0 = 0.9$.
Furthermore, we can differentiate φ in the surrounding $B_{0.1}(x^*) \cap \mathcal{F}_p$ with $\nabla\varphi(x^*) = 1$ and therefore can find the MFCQ-vector $v = 1$.
Next, we focus on the subset scheme which is defined here as the union of $k \in \mathbb{N}$ points

$$\Phi_k : X \times 2^T \to 2^T, (x,S) \to T_k \coloneqq \bigcup_{n=1}^{k} \left\{1 - \frac{1}{n}\right\}.$$

This subset scheme is by definition an increasing discretization scheme and does fulfill $t^*(x,\xi) = 1 \in \operatorname{cl}(S)$ *with* $S = \lim_{k\to\infty} T_k$ *for any* $x \in [0,1], \xi \in [0,1]$. *With this fixed subset scheme, we can represent*

$$\Omega(x, T_k) = \left[x, \min\left\{2x + 0.9 + \frac{1}{k}, 1\right\}\right]$$

for any fixed decision $x \in [0,1]$.
As $x^* = 0$ *is not only the minimizer of the probust optimization problem, but also the minimizer of the optimization problem* $\min_{x\in[0,1]} x$ *without additional constraint, we know that the probust subset schemes will find* $\underline{x}_k = x^* = 0$ *for all iterations* $k \in \mathbb{N}$. *Therefore, we know that for fixed iteration* $k \in \mathbb{N}$ *we can represent*

$$\Omega(\underline{x}_k, T_k) = \left[0, \min\left\{0.9 + \frac{1}{k}, 1\right\}\right].$$

Now, we consider the different set-approximation schemes (4.1) *to* (4.3).

1. $D_k(x,\delta) = \Omega(y, T_k)$: *Considering the maximum constraint* (3.4) *of the inner set-approximation problem and interpreting it as a set-inclusion constraint, we have to satisfy*

$$\Omega(y, T_k) \subseteq \Omega(x, T) \Leftrightarrow \left[y, 2y + 0.9 + \frac{1}{k}\right] \subseteq [x, 2x + 0.9].$$

for some $(x,y) \in [0,1]^2$. *This inclusion implies*

$$\begin{aligned} &x \leq y \textit{ and} \\ &2y + 0.9 + \frac{1}{k} \leq 2x + 0.9 \Leftrightarrow y \leq x - \frac{1}{2k}, \end{aligned}$$

which is a contradiction for all $k \in \mathbb{N}$. *This means that we cannot find a feasible solution for any inner set-approximation problem.*

2. $D_k(x,\delta) = \Omega(\underline{x}_k, T_k)$: *Since this set is a special case of* 1. *with fixed* $y = x_k$ *the same arguments as for the last design imply for the maximum constraint* (3.4) *of the inner set-approximation problem*

$$\Omega(\underline{x}_k, T_k) \subseteq \Omega(x, T) \Leftrightarrow \left[0, 0.9 + \frac{1}{k}\right] \subseteq [x, 2x + 0.9]$$

and therefore

$$\begin{aligned} &x \leq 0 \textit{ and} \\ &0.9 + \frac{1}{k} \leq 2x + 0.9 \Leftrightarrow \frac{1}{2k} \leq x, \end{aligned}$$

which is also a contradiction for each $k \in \mathbb{N}$.

3. $D_k(x,\delta) = \Omega_r(y,T_k)$: *In contrast to the last two examples, these design lead to a sequence of set-approximation problems such that the choice* $x_k = \frac{1}{4k}, y_k = 0, r_k = \frac{1}{2k}$ *is feasible for all problems starting from iteration* $k \geq \frac{1}{2(1-p)} = 5$ *as we can check*

$$\Omega_{r_k}(y_k,T_k) \subseteq \Omega(x_k,T) \Leftrightarrow \left[y_k + r_k, 2y_k + 0.9 + \frac{1}{k} - r_k\right] \subseteq [x_k, 2x_k + 0.9]$$
$$\Leftrightarrow \left[\frac{1}{2k}, 0.9 + \frac{1}{2k}\right] \subseteq \left[\frac{1}{4k}, 0.9 + \frac{1}{2k}\right]$$

and

$$\mathbb{P}(\Omega_{r_k}(y_k,T_k)) = \mathbb{P}\left(\left[\frac{1}{2k}, 0.9 + \frac{1}{2k}\right]\right) = 0.9.$$

Because the sequence $x_k = \frac{1}{4k}$ *converges towards the minimizer of the original probust optimization problem* $x^* = 0$*, this set-approximation scheme seems to be more appropriate for this problem instance. As it covers the first case by fixing* $r = 0$ *we expect it to be the most flexible design to find a solution.*

This example shows that set-approximation schemes with information from probust subset schemes do not converge in general. To be able to approximate the set $\Omega(x^*)$ we should start with generic designs like $\Omega_r(y,T_k)$ and get more specific by choosing $\Omega(y,T_k)$ or $\Omega(x_k,T_k)$ if the first design leads to an solvable problem. If the design $\Omega_r(y,T_k)$ can always be used given the setting of Theorem 3.3.3 is not clear so far.
One structural problem in general is that increasing subset schemes imply $T_k \subseteq T_{k+1}$ for all $k \in \mathbb{N}$ and therefore a decreasing sequence $\Omega(x,T_{k+1}) \subseteq \Omega(x,T_k)$ of the corresponding designs for any $x \in X$. To be able to use Theorem 3.3.3, we have to ensure an increasing set of designs $\mathcal{D}_k \subseteq \mathcal{D}_{k+1}$ for all $k \in \mathbb{N}$. Consequently, we have to find for each $x \in X, k \in \mathbb{N}$ some $y \in X$ such that $D_k(x,T_k) = D_{k+1}(y,T_{k+1})$ which implies $\Omega(x,T_k) = \Omega(y,T_{k+1})$. Since the set of feasible realizations changes its structure by adding discretization points, this conditions seems hard to fulfill in general.

To find sufficient conditions to guarantee the convergence of inner and outer approximations based on pieces of information generated by probust subset methods is one direction that future work can go.
Nevertheless, we focus on the sandwiching algorithm 6, where we specify $D_k = \Omega(\underline{x}_k,T_k)$ for all $k \in \mathbb{N}$ for the upcoming applications. As explained, this approach allows us to calculate an upper bound for our objective function fast, even if it does not converge towards any accumulation point of the iterates created by a probust subset scheme. Keeping this point in mind, we have to check in every iteration if the solution $\overline{x}_k$ is feasible w.r.t. the inner set-approximation. The corresponding algorithm which uses information of the probust subset scheme in the set-approximation scheme is formulated as:

Algorithm 7 Exchange sandwiching algorithm for standard probust optimization problems

```
Inputs:
    Probust optimization problem instance SPP, precision $\epsilon > 0$,
    discretization scheme $\Phi$, initial decision $x_0 \in X$,
    initial discretization $T_0 \subseteq T$
Initialize:
    $k := 0$
do
    $\underline{x}_{k+1} \leftarrow \arg\min_{x \in X} f(x)$ s.t. $\varphi(x, T_k) \geq p$
    $T_{k+1} \leftarrow \Phi_k(\underline{x}_{k+1}, T_k)$
    $\overline{x}_{k+1} \leftarrow \arg\min_{x \in X} f(x)$ s.t. $\max_{\xi \in \Omega(\underline{x}_{k+1}, T_k), t \in T} g(x, \xi, t) \leq 0$
    $k \leftarrow k + 1$
while $f(\overline{x}_k) - f(\underline{x}_k) \geq \epsilon$
Results:
    Sequence $(\underline{x}_k, \overline{x}_k, T_k)_{k \in \mathbb{N}}$
```

Before testing this sandwiching algorithm on applications in the Part II of this thesis, we comment on the idea of using information from set-approximation schemes in the probust subset scheme.

4.2 Set-approximation schemes using probust subset information

In this section, we comment on using information generated by set-approximation schemes to define a probust subset scheme. We have seen that probust subset schemes define outer approximations of the corresponding probust optimization problem (see Lemma 2.2.2), but cannot guarantee to find a solution that is feasible w.r.t. the original problem in general.

As inner set-approximation methods do find such feasible decisions, we focus on how we can make this useful for the probust subset schemes.

We assume that we use a set-approximation scheme Ψ satisfying the assumptions of Theorem 3.3.2. As the theorem guarantees that we find the "shape" of the set of feasible realizations $\Omega(\overline{x})$, where $\overline{x}$ is the accumulation points $\overline{x} \in X$ of the sequence $(\overline{x}_k)_{k \in \mathbb{N}}$ generated by the inner set-approximation problems, we hope that the corresponding probust subset schemes do also converge.

We do not analyze the convergence of this approach here, but comment on the following ways to use information from inner set-approximation problems in probust subset schemes:

1. We could use the information $(\overline{x}_k, \delta_k)$ to calculate subsets T_k of T by defining probust subset schemes Φ. This approach is troublesome because the set $D(\overline{x}_k, \delta_k)$ is independent w.r.t. $t \in T$ in general and therefore, it is unclear how to calculate scenarios given $D(\overline{x}_k, \delta_k)$. On the contrary, we can try to use the iterates $\overline{x}_k$ as fixed arguments for adaptive subset schemes such as the HLAD or LLAD.

2. We could use the information $(\overline{x}_k, \delta_k)$ to define a new optimization problem to determine $\underline{x}_k$. In comparison to the inner set-approximation from Chapter 3, we think about the following *outer set-approximation problem*, where we are searching for supersets of $\Omega(x), x \in X$:

$$\begin{aligned} \min_{x \in X} f(x) \text{ s.t. } & \mathbb{P}(D_k(x, \delta)) \geq p \ \forall \delta \in \Delta, \\ & \Omega(x) \subseteq D_k(x, \delta) \ \forall \delta \in \Delta. \end{aligned}$$

   In this case, it is not enough to just find one design parameter $\delta \in \Delta$ that satisfies $\mathbb{P}(D(x, \delta)) \geq p$, but we have to guarantee this condition for all possible $\delta \in \Delta$. Consequently, we cannot use $\delta \in \Delta$ as a decision variable and the probability constraint gets a necessary condition to guarantee $\mathbb{P}(\Omega(x)) \geq p$.
   Because this condition is hard to check, we simplify the outer set-approximation problem by reducing the design space to just one element $\Delta_k = \{\overline{\delta}\}$ for all $k \in \mathbb{N}$ defining an unique design $\overline{D}_k$. The corresponding optimization problem then reads

$$\begin{aligned} \min_{x \in X} f(x) \text{ s.t. } & \mathbb{P}(\overline{D}_k) \geq p, \\ & \Omega(x) \subseteq \overline{D}_k. \end{aligned}$$

   Here the questions arise how to fix the set $\overline{D}_k$ and how to solve the generalized semi-infinite constraint induced by $\Omega(x) \subseteq \overline{D}_k$ efficiently.

So far, we do not see any helpful combination of the parameters $(\overline{x}_k, \delta_k)_{k \in \mathbb{N}}$ from a set-approximation scheme with the iterates of a probust subset scheme $(\underline{x}_k, T_k)_{k \in \mathbb{N}}$.
We think, that this way of combining information is not helpful in general because we start with a set-approximation scheme to generate subsets $S \subseteq T$ that describe the set of feasible realizations for a given decision. As the set-approximation scheme converges towards a solution of the original probust optimization problem by assumption, we know that the sequence $(D_k(\overline{x}_k, \delta_k))_{k \in \mathbb{N}}$ already approximates this set of feasible realizations and therefore the calculation of $S \subseteq T$ doubles the effort.
Furthermore, we do not see any subproblem in this procedure that can be solved easily to generate additional information while running a set-approximation scheme.
Therefore, we do not use any of the ideas above numerically in Part II of this thesis.

Overview Part I

As we reached the end of the theory part of this thesis, we want to recapitulate what we have achieved within the Chapters 2 - 4.
We do this, on the on hand, by shortly mentioning the main theorems in these chapters and then visualize the newly defined solution approach for standard probust optimization problems by Figure 4.1 which depicts an updated workflow how to handle a probust optimization problem.

In Chapter 2, we considered discretization schemes inspired by the literature of semi-infinite optimization to replace the set of scenarios T of infinite cardinality by a sequence of finite subsets $(T_k)_{k\in\mathbb{N}}, T_k \subseteq T$ for all $k \in \mathbb{N}$. A condition to check if these sets are appropriate to replace T is given by the candidate-condition (see Definition (2.1.9)) at the current iterate $x_k \in X$ which we deduced from analyzing probust terms.
Using this condition, we define discretization schemes such as a uniform discretization scheme with vanishing grid sizes or two probust variants of the adaptive discretization scheme from Blankenship and Falk whose iterates converge to a solution of the original probust optimization problem.
The main theorem to ensure convergence of such discretization schemes is Theorem 2.2.3. Since the stated Algorithm 3 is an iterative one without explicit stopping criterion, we are interested in finding feasible iterates for a probust optimization problem.
Rethinking chance constrained optimization and connecting it to design-centering problems (that are introduced in the next chapter) in Chapter 3, we defined the inner set-approximation problem (see Definition 3.2.1) whose optimal value is an upper bound for the optimal value of the original problem. Interpreting standard probust optimization problems as chance constraints with a special structure of the inner function, we could transfer the approximation and convergence results (see Proposition 3.2.2, Theorem 3.3.2 and Theorem 3.3.3) found for the chance constrained problem class to standard probust optimization problems.
In Chapter 4, we discussed how we can combine both solution approaches efficiently. Considering several methods where information of one solution approach are used in the other one and an example problem to test these algorithms are given. In the end, we came up with one specific combination of the solution approaches stated as Algorithm 7. This algorithm uses the output of a probust subset scheme to reduce the inner-set approximation problem to a mere SIP. As we hope that this problem can be solved within a reasonable numerical effort, we will use this algorithm for the following applications. As we have not fixed the probust subset scheme for this algorithm, we can specify it according to the structure of the considered problem structure.

The workflow of how to handle a standard probust optimization problem with the solution approaches defined in Part I of this thesis can be visualized as follows:

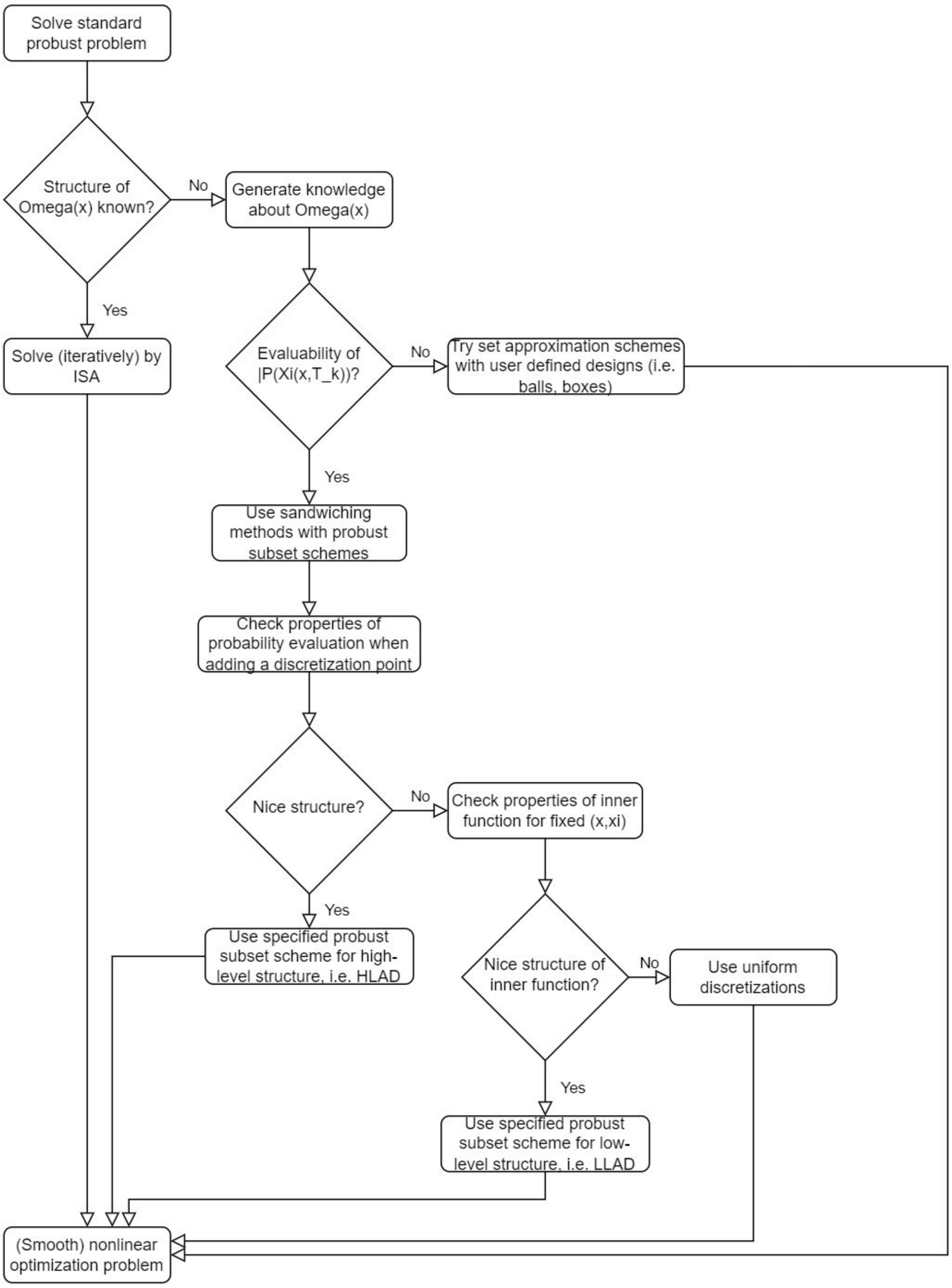

Solve standard probust problem
Structure of Omega(x) known?
No
Generate knowledge about Omega(x)
Yes
Solve (iteratively) by ISA
Evaluability of |P(Xi(x,T_k))?
No
Try set approximation schemes with user defined designs (i.e. balls, boxes)
Yes
Use sandwiching methods with probust subset schemes
Check properties of probability evaluation when adding a discretization point
Nice structure?
No
Check properties of inner function for fixed (x,xi)
Yes
Use specified probust subset scheme for high-level structure, i.e. HLAD
Nice structure of inner function?
No
Use uniform discretizations
Yes
Use specified probust subset scheme for low-level structure, i.e. LLAD
(Smooth) nonlinear optimization problem

Figure 4.1: Workflow how to solve probust optimization problems defined by Part I of this thesis

This workflow is meant to start, where Figure 1.5 in Chapter 2 needed problem specific solution to "Solve standard probust optimization problem".

We first ask ourselves if we know how the set of feasible realizations is structured.

If we can describe this structure by functions as in Proposition 3.1.7, we should consider an inner set-approximation problem to reformulate this problem and solve it. Here, we need to evaluate the probability constraint (3.3) which we assume is possible for the specified structure of the set of feasible realizations. Furthermore, we have to evaluate the set-inclusion constraint (3.4) which we assume to be reducible to a semi-infinite constraint by the given structure of the sets of feasible realizations using an appropriate transformation of the uncertainty set like in [72]. Consequently, the problem should be solvable within reasonable numerical effort.

If we do not know the structure of sets of the feasible realizations, we have to generate knowledge about these sets. Either we just test set-approximation schemes which imply a lot of numerical effort or we try to use the problem structure given by the probust optimization problem evaluating approximations of the set of feasible realizations $\Omega(x, S)$ for $x \in X$ and $S \subseteq T$. If we can evaluate the probability of such sets numerical efficiently, we can consider the sandwiching scheme given by Algorithm 7. Here, we have to specify the probust subset scheme that we want to use.

With the experience of the second part of this thesis, we recommend to test if we can get additional analytical insight from this scenario-wise solution approach to specify "important" discretization points. If so, we can use these information to define an specified subset scheme that needs a few well-chose discretization points such as the (MI-)HLAD in Section 2.3. As we just take a few discretization points, the calculation of $\underline{x}_k, \overline{x}_k$ in each iteration will be relatively fast.

If we have no additional insight into the importance of scenarios, then the search for such scenarios might take too much time or lead to a break down of the used numerical tool. Consequently, we have to use explorative discretization schemes such as the uniform discretization scheme to find important discretization points for all decisions $x \in X$ (see Lemma 2.3.1). Such methods define more discretization points and therefore increase the running time when calculating iterates $\underline{x}_k, \overline{x}_k$, but they find discretization points more reliably. If the structure of the inner function g is useable for an optimization w.r.t. $t \in T$- e.g. a concave function w.r.t. $t \in T$- then we recommend choosing the LLAD over choosing a uniform discretization approach.

With the information from explorative discretization schemes we might improve our understanding of important scenarios and with them the understanding of the structure of the sets of feasible realizations.

This concludes the workflow depicted in Figure 4.1 of handling standard probust optimization problems defined by the introduced solution approaches.

Our next goal is to use these solution approaches to get solutions for realistic probust optimization problems and thereby gain insights about the basic problem classes.

Part II

Applications

5 Stochastic design-centering problems

In this chapter we give a short introduction to (stochastic) design-centering problems and how they are connected to semi-infinite and probust optimization problems.
After that we define three problem instances in Section 5.1 which motivate us to modify the probust subset schemes introduced in Section 2.3 to fit their inner function structure in Section 5.2. As we can find analytical solutions of these three stochastic design-centering problem instances, we use them to compare the probust solution approaches introduced in the first part of this thesis concerning precision of the solution and running time in Section 5.3. We also gather numerical experience handling these problems that can be useful for the remaining applications in the second part of this thesis.

A design-centering problem has the following form

$$\text{DC}: \max_{x \in X} vol(D(x)) \text{ s.t. } D(x) \subseteq C(x),$$

where $C(x) \subseteq \mathbb{R}^m$ is a so called container in which the parameter-depending design $D(x) \subseteq \mathbb{R}^m$ should lie within for some $x \in X$. As we can influence the form of the design and the container by the parameter $x \in X$, we want to maximize the volume vol of the design within the container.
If we assume that the container can be described by finite many inequality constraints $g_i : X \times \mathbb{R}^q \to \mathbb{R}, i \in I, |I| < \infty$ as

$$C(x) = \{y \in \mathbb{R}^q \mid g_i(x, y) \leq 0 \ \forall i \in I\},$$

we can reformulate a DC as a generalized semi-infinite optimization problem

$$\text{DC-SIP}: \min_{x \in X} -vol(D(x)) \text{ s.t. } g_i(x, y) \leq 0 \ \forall y \in D(x), i \in I.$$

This kind of problem is handled in several works, e.g., in Section 2.4 of Oliver Stein's book [78] and paper's from Stein [79], Winterfeld [86] and Harwood et al. [39].

Inspired by DCs, we extend the problem description by a random variable Z that can be interpreted as a random disturbance when adjusting the design into the container.
We call the following extended version of the problem stochastic design-centering problem (SDC) and can formulate it as

$$\text{SDC}: \max_{x \in X} \mathbb{E}(vol(D(x, \xi))) \text{ s.t. } \mathbb{P}(D(x, \xi) \subseteq C(x, \xi)) \geq p.$$

If we again assume that the container can be described by finite many inequality constraints $g_i : X \times \Xi \times \mathbb{R}^q \to \mathbb{R}, i \in I, |I| < \infty$ as

$$C(x, \xi) = \{y \in \mathbb{R}^q \mid g_i(x, \xi, y) \leq 0 \; \forall i \in I\},$$

we can reformulate a SDC as a generalized probust optimization problem

$$\text{SDC-GPP} : \min_{x \in X} -\mathbb{E}(vol(D(x, \xi))) \text{ s.t. } \mathbb{P}(g_i(x, \xi, y) \leq 0 \; \forall y \in D(x, \xi), i \in I) \geq p.$$

One simple example is that we want to maximize the length of the interval $I(x) := [0, x]$ within the container $C := [0, 1]$, where $x \in [0, 1]$. When adjusting this interval I into C we have to consider a uncertain influence $Z \sim \mathcal{U}([0, 1])$ that influences the positioning of $I(x)$ by $D(x, \xi) = I(x) + [0, \xi] = [0, x + \xi]$.
Given a probability $p \in [0, 1]$ with that we should guarantee that $D(x, \xi)$ lies within $[0, 1]$, we can solve the induced probust optimization problem.
Since the solution of this probust optimization problem can be calculated as $x_p^* = 1 - p$, we can interpret the difference between this minimizer and $x^* = 1$ - as the minimizer corresponding to the (deterministic) constraint $I(x) \subseteq C$ - as a safety buffer to be able to react to the influence of the realizations of the random vector Z. How big this safety buffer is depends mainly on $p \in [0, 1]$ and the used probability distribution.
In the following section we define some more complex examples of stochastic design-centering problems.

5.1 Problem instances

The focus of this section is to define three examples of stochastic design-centering problems with different structural properties. We calculate their optimal solutions before we compare the performance of different numerical solving strategies inspired by the schemes introduced in Chapter 2 - 4.
We handle the single stochastic design-centering problems in the following way:
First, we introduce the container and the design which we focus on and represent them by inequality constraints. The stochastic design-centering problem (SDC) induced by the sets can then be rewritten as a generalized probust optimization problem (SDC-GPP). Afterwards, we introduce transformations of the uncertainty sets which allow us to reduce the generalized probust optimization problem to a standard probust optimization problem (SDC-SPP). Then, we can use statements from Chapter 1 to ensure an unique solution for each problem instance. Finally, we calculate these solutions analytically.

Example: Circle in uncertain triangle

Problem formulation: We consider the stochastic design-centering problem of putting an area maximizing circle given by the design

$$D(x) = B_{x_1}\left(\begin{pmatrix} x_2 \\ x_3 \end{pmatrix}\right) = \{y \in \mathbb{R}^2 \mid (y_1 - x_2)^2 + (y_2 - x_3)^2 \leq x_1^2\}$$

with $x \in X := \mathbb{R}_{\geq 0} \times \mathbb{R}^2$ into an uncertain triangle interpreted as the container that is described by the points

$$\begin{aligned} P_1 &= (-1, Z), \\ P_2 &= (1, Z), \\ P_3 &= (0, Z+1), Z \sim \mathcal{N}(0, 0.2). \end{aligned}$$

Consequently, we can describe the triangle for a fixed $\xi \in \mathbb{R}$ as

$$\begin{aligned} C(\xi) = \{y \in \mathbb{R}^2 \mid g_1(\xi, y) \leq 0, g_2(\xi, y) \leq 0, g_3(\xi, y) \leq 0\}&, \\ \text{where } g_1(\xi, y) = \xi - y_2&, \\ g_2(\xi, y) = y_2 + y_1 - \xi - 1 \text{ and}& \\ g_3(\xi, y) = y_2 - y_1 - \xi - 1.& \end{aligned}$$

Fixing a threshold parameter $p := 0.9$ leads to the stochastic design-centering problem

$$\text{SDC} : \max_{x \in X} \pi x_1^2 \text{ s.t. } \mathbb{P}(D(x) \subseteq C(\xi)) \geq p.$$

Problem reformulation (GPP): Before analyzing this problem, we have to reformulate its constraints in a way that we can handle.
Therefore, we rewrite SDC as a generalized probust optimization problem as explained in the beginning of this chapter:

$$\text{SDC-GPP} : \min_{x \in X} -\pi x_1^2 \text{ s.t. } \mathbb{P}\begin{pmatrix} g_1(\xi, y) \leq 0 \ \forall y \in D(x), \\ g_2(\xi, y) \leq 0 \ \forall y \in D(x), \\ g_3(\xi, y) \leq 0 \ \forall y \in D(x) \end{pmatrix} \geq 0.9$$

Problem reformulation (SPP): Next we use the following transformation to the reference set $\hat{T} := B_1(0)$ to simplify the set-dependency within the probability evaluation

$$\mathcal{T}_{\hat{T}} : X \times \Xi \times \hat{T} \to \mathbb{R}^2, (x, \xi, z) \to x_1 \cdot z + \begin{pmatrix} x_2 \\ x_3 \end{pmatrix}.$$

As $\mathcal{T}_{\hat{T}}(x, \xi, \cdot)$ is surjective for all $x \in X := \mathbb{R}_{\geq 0} \times \mathbb{R}^2, \xi \in \Xi := \mathbb{R}$ and even a homeomorphism between $\hat{T} = B_1(0)$ and $D(x)$ if $x_1 \neq 0$, we can reformulate the inner functions of the original problem as

$$\begin{aligned} \tilde{g}_1(x, \xi, z) &= g_1(\xi, \mathcal{T}_{\hat{T}}(x, \xi, z)) = \xi - x_1 z_2 - x_3, \\ \tilde{g}_2(x, \xi, z) &= g_2(\xi, \mathcal{T}_{\hat{T}}(x, \xi, z)) = x_1 z_2 + x_3 + x_1 z_1 + x_2 - \xi - 1, \\ \tilde{g}_3(x, \xi, z) &= g_3(\xi, \mathcal{T}_{\hat{T}}(x, \xi, z)) = x_1 z_2 + x_3 - x_1 z_1 - x_2 - \xi - 1. \end{aligned}$$

This leads to the following standard probust optimization problem:

$$\text{SDC-SPP}: \min_{x_1\geq 0, x_2, x_3 \in \mathbb{R}} -\pi x_1^2 \text{ s.t. } \mathbb{P}\begin{pmatrix} \tilde{g}_1(x,\xi,z) \leq 0 \ \forall z \in B_1(0), \\ \tilde{g}_2(x,\xi,z) \leq 0 \ \forall z \in B_1(0), \\ \tilde{g}_3(x,\xi,z) \leq 0 \ \forall z \in B_1(0) \end{pmatrix} \geq 0.9$$

Analysis (solution existence): After reformulating the optimization problem into a form we can analyze, we use the theory introduced in Section 1.1.
Because $\tilde{g}_i$ is continuous w.r.t. $(x,\xi,z) \in X \times \Xi \times \hat{T}$, we know by Proposition 1.1.6 that $\varphi : X \to [0,1]$ is upper semi-continuous. Consequently the feasible set of SDC-SPP is closed.
It is also compact, as we can reduce the search space $X = \mathbb{R}_{\geq 0} \times \mathbb{R}^2$ to $\tilde{X} = [0,1] \times [-1,1]^2$. This reduction is possible as the radius of the circle is not allowed to be bigger than the height of the triangles $C(\xi), \xi \in \Xi$, which justifies $x_1 \in [0,1]$. Furthermore, the midpoint $(x_2, x_3) \in \mathbb{R}^2$ has to be contained within some triangle which explains $x_2 \in [-1,1]$. Last, but not least $x_3 \in [-1,1]$ is guaranteed because any decision $x \in X$ with $|x_3| \geq 1$ cannot be feasible as we will see next:
Any two containers fulfill $C(\xi_1) \cap C(\xi_2) = \emptyset$, if $|\xi_1 - \xi_2| > 1$. Consequently, no design can lie within both such containers. If we assume $|x_3| > 1$, then we can estimate

$$\varphi(x) \leq \mathbb{P}([|x_3| - 1, |x_3|]) \leq \mathbb{P}([0,1]) \leq \frac{1}{2} < p.$$

Therefore, the decision is infeasible as claimed.
All together, we can reduce the search space X to the compact space $\tilde{X} = [0,1] \times [-1,1]^2$.
Next, we have to guarantee that the feasible set is not empty. Therefore, we concentrate on estimating $p_{\max} \in [0,1]$:
We have already observed that $C(\xi_1) \cap C(\xi_2) = \emptyset$ for all $|\xi_1 - \xi_2| > 1, \xi_1, \xi_2 \in \Xi$. This means that the set of feasible realizations for a fixed decision is contained within in interval of length 1.
Fixing the decision $x_0 = \left(0, 0, \frac{1}{2}\right)$, we know that the set of feasible realizations is given by $\Omega(x_0) = \left[-\frac{1}{2}, \frac{1}{2}\right]$ by plotting the corresponding design. This leads to the estimation

$$p_{\max} = \max_{x \in X} \varphi(x) \geq \varphi(x_0) = \mathbb{P}\left(\left[-\frac{1}{2}, \frac{1}{2}\right]\right) = F\left(\frac{5}{2}\right) - F\left(-\frac{5}{2}\right) \approx 0.9876.$$

With $p = 0.9 < p_{\max}$ this implies that the feasible set is not empty and thus SDC-SPP has a well-defined solution.
Analysis (convex feasible set): Furthermore, we know by Proposition 1.1.8 that $\varphi : X \to [0,1]$ is a log-concave function and therefore the feasible set of SDC-SPP is convex. The assumptions of Proposition 1.1.8 hold because normal distributions are log-concave, the maximum of quasi-convex functions is quasi-convex and because linear functions are quasi-convex. We cannot argue that the objective function is strictly convex w.r.t. $x \in X$, but the next solution step shows that the minimizer of SDC-SPP is unique.
Analysis (solution calculation): Because we know that we can find a solution of the optimization problem SDC-SPP, we can try to calculate it:
As we are missing a sufficient condition to check if a given point $x^* \in X$ is a minimizer of a probust optimization problem, we have to construct the solution.

By symmetry of the triangle and because changes in the realization $\xi \in \mathbb{R}$ just influence the second coordinate of the circle, we know that the optimal decision $x^* = (x_1^*, x_2^*, x_3^*)$ satisfies $x_2^* = 0$. Now we go through the solving steps introduced in Example 1.1.9.
We start with calculating $\max_{z \in \hat{T}} \tilde{g}(x, \xi, z) = \max_{z \in \hat{T}} \max_{i=1,2,3} \tilde{g}_i(x, \xi, z)$, where we go through the single constraints individually

$$\max_{z \in B_1(0)} \tilde{g}_1(x, \xi, z) = \xi + x_1 - x_3 \text{ with } z_1^* = (0, -1),$$

$$\max_{z \in B_1(0)} \tilde{g}_2(x, \xi, z) = \sqrt{2}x_1 + x_2 + x_3 - \xi - 1 \text{ with } z_2^* = \frac{\sqrt{2}}{2}(1, 1) \text{ and}$$

$$\max_{z \in B_1(0)} \tilde{g}_3(x, \xi, z) = \sqrt{2}x_1 - x_2 + x_3 - \xi - 1 \text{ with } z_3^* = \frac{\sqrt{2}}{2}(-1, 1).$$

Please note that the worst-case scenarios $z_i^*, i = 1, 2, 3$ do not depend on $x \in X$ or $\xi \in \Xi$. Now, we can reformulate the probust constraint in a second step by a joint chance constrained:

$$\begin{aligned}
\mathbb{P}\begin{pmatrix} \tilde{g}_1(x, \xi, z) \leq 0 \ \forall z \in B_1(0), \\ \tilde{g}_2(x, \xi, z) \leq 0 \ \forall z \in B_1(0), \\ \tilde{g}_3(x, \xi, z) \leq 0 \ \forall z \in B_1(0) \end{pmatrix} &= \mathbb{P}\begin{pmatrix} \xi + x_1 - x_3 \leq 0, \\ \sqrt{2}x_1 + x_2 + x_3 - \xi - 1 \leq 0, \\ \sqrt{2}x_1 - x_2 + x_3 - \xi - 1 \leq 0 \end{pmatrix} \\
&= \mathbb{P}\begin{pmatrix} \xi \leq x_3 - x_1, \\ \sqrt{2}x_1 + x_2 + x_3 - 1 \leq \xi, \\ \sqrt{2}x_1 - x_2 + x_3 - 1 \leq \xi \end{pmatrix} \\
&= \mathbb{P}([\underbrace{\sqrt{2}x_1 + \max\{x_2, -x_2\} + x_3 - 1}_{=:\underline{\xi}(x)}, \underbrace{x_3 - x_1}_{=:\overline{\xi}(x)}]) \\
&= \mathbb{P}([\underline{\xi}(x), \overline{\xi}(x)] \qquad (5.1)
\end{aligned}$$

In a third and last step we want to choose an optimal decision variable $x^* \in X$ that satisfies the joint chance constrained:
We take note that the interval $[\underline{\xi}(x), \overline{\xi}(x)]$ is monotonically decreasing w.r.t. $\subseteq$ for increasing $x_1 \geq 0$. Furthermore, we notice that the decision variables $x_2, x_3 \in \mathbb{R}$ do not influence the objective value directly. They are used to guarantee the feasibility of the decision variable $x_1 \geq 0$ which should be chosen as big as possible as the objective function is monotonically decreasing w.r.t. $x_1 \geq 0$. Therefore, an optimal decision x^* does not only imply $x_2^* = 0$ as explained above, but also chooses x_3^* in a way such that the probability is as high as possible. Due to the normal distribution in this example this means that x_3^* guarantees the symmetry of the set of feasible realizations $\Omega(x^*)$ around the expected value $\mu = 0$ which induces the explicit form $\Omega(x^*) = [\underline{\xi}(x^*), \overline{\xi}(x^*)]$. Consequently, we can derive the following necessary condition for an optimal solution $x^* = (x_1^*, x_2^*, x_3^*)$ with $x_2^* = 0$:

$$\begin{aligned}
&-\underline{\xi}(x^*) = \overline{\xi}(x^*) \\
\Leftrightarrow &-(\sqrt{2}x_1^* + x_3^* - 1) = x_3^* - x_1^* \\
\Leftrightarrow &x_3^* = \frac{1 - \sqrt{2}}{2}x_1^* + \frac{1}{2}
\end{aligned}$$

Inserting this condition into the bounds $\underline{\xi}(x^*), \overline{\xi}(x^*)$ we observe that these are monotonically decreasing w.r.t. x_1^*. Consequently, the probust constraint is active for the optimal choice x^* which leads to an equality constraint that defines the value of x_1^* by

$$\begin{aligned} p &= \mathbb{P}([\underline{\xi}(x^*), \overline{\xi}(x^*)]) \\ &= \mathbb{P}\left([-\overline{\xi}(x^*), \overline{\xi}(x^*)]\right) \\ &= 2\mathbb{P}\left([0, \overline{\xi}(x^*)]\right) \\ &= 2\left(F\left(\frac{\overline{\xi}(x^*)}{\sigma}\right) - F(0)\right) \\ &= 2\left(F\left(\frac{x_3^* - x_1^*}{\sigma}\right) - \frac{1}{2}\right) \\ &= 2F\left(\frac{5}{2}(1 - (1+\sqrt{2})x_1^*)\right) - 1, \end{aligned}$$

where F denotes the cumulative distribution function of a standard normal distributed random variable. Remembering $p = 0.9$, the last equality is equivalent to

$$x_1^* = \frac{1 - 0.4F^{-1}\left(\frac{1+p}{2}\right)}{1+\sqrt{2}} \approx 0.1417.$$

With this we can calculate the (unique) optimal solution via

$$\begin{aligned} x_3^* &= \frac{1 - \sqrt{2}x_1^*}{2} + \frac{1}{2} \approx 0.4707, \\ x^* &\approx (0.1417, 0, 0.4707) \text{ and} \\ f(x^*) &\approx -6.308 \cdot 10^{-2}. \end{aligned}$$

Example: Ellipse in uncertain triangle

Problem formulation: We consider the stochastic design-centering problem of putting an area maximizing ellipse given by the design

$$D(x) = A(x)B_1(0) + b(x) = \{y \in \mathbb{R}^2 \mid \exists z \in B_1(0) : y = A(x)z + b(x)\}$$

with $A(x) = \begin{pmatrix} x_1 & x_3 \\ 0 & x_2 \end{pmatrix}, b(x) = \begin{pmatrix} x_4 \\ x_5 \end{pmatrix}, x \in X := \mathbb{R}^2_{\geq 0} \times \mathbb{R}^3$ into an uncertain triangle interpreted as the container that is described by the points

$$\begin{aligned} &P_1 = (-\sqrt{3}, -1), \\ &P_2 = (\sqrt{3}, -1), \\ &P_3 = (0, 2 + Z), Z \sim \mathcal{N}(0,1). \end{aligned}$$

Please note that this triangle is degenerated for the realization $\xi = -3$ what happens with probability $\mathbb{P}(\xi = -3) = 0$.
We can describe the triangle for a fixed realization $\xi \in \mathbb{R}$ as

$$\begin{aligned} C(\xi) = \{y \in \mathbb{R}^2 \mid g_1(\xi,y) \leq 0, g_2(\xi,y) \leq 0, g_3(\xi,y) \leq 0\}, \\ \text{where } g_1(\xi,y) = \operatorname{sign}(3+\xi)(-1-y_2), \\ g_2(\xi,y) = \operatorname{sign}(3+\xi)\left(\sqrt{3}y_2 + (3+\xi)y_1 + \sqrt{3}(2+\xi)\right) \text{ and} \\ g_3(\xi,y) = \operatorname{sign}(3+\xi)\left(\sqrt{3}y_2 - (3+\xi)y_1 + \sqrt{3}\right). \end{aligned}$$

Here the sign-function is necessary because for $\xi < -3$ the described triangle flips over. Nevertheless, we will see that this case is to improbable to consider and therefore we can ignore the sign-function in the model description.
With $p := 0.9$ the stochastic design-centering problem can be written as

$$\text{SDC} : \max_{x \in X} \pi x_1 x_2 \text{ s.t. } \mathbb{P}(D(x) \subseteq C(\xi)) \geq p.$$

Problem reformulation (GPP): Before analyzing this problem, we have to reformulate its constraints in a way that we can handle them.
Therefore, we rewrite SDC as a generalized probust optimization problem as explained in the beginning of this chapter:

$$\text{SDC-GPP} : \min_{x \in X} -\pi x_1 x_2 \text{ s.t. } \mathbb{P}\begin{pmatrix} g_1(\xi,y) \leq 0 \ \forall y \in D(x), \\ g_2(\xi,y) \leq 0 \ \forall y \in D(x), \\ g_3(\xi,y) \leq 0 \ \forall y \in D(x) \end{pmatrix} \geq 0.9$$

Problem reformulation (SPP): Next we use the following transformation to the reference set $\hat{T} := B_1(0)$ to simplify the set-dependency within the probability evaluation

$$\mathcal{T}_{\hat{T}} : X \times \Xi \times \hat{T} \to \mathbb{R}^2, (x, \xi, z) \to A(x)z + b(x).$$

As $\mathcal{T}_{\hat{T}}(x, \xi, \cdot)$ is surjective for all $x \in X, \xi \in \Xi := \mathbb{R}$ and even a homeomorphism between $\hat{T} = B_1(0)$ and $D(x)$ if $x_1 \cdot x_2 \neq 0$, we can reformulate the inner functions of the original problem as

$$\tilde{g}_1(x, \xi, z) = g_1(\xi, \mathcal{T}_{\hat{T}}(x, \xi, z)) = -1 - (x_2 z_2 + x_5),$$
$$\tilde{g}_2(x, \xi, z) = g_2(\xi, \mathcal{T}_{\hat{T}}(x, \xi, z)) = \sqrt{3}(x_2 z_2 + x_5) + (3 + \xi)(x_1 z_1 + x_3 z_2 + x_4) + \sqrt{3}(2 + \xi),$$
$$\tilde{g}_3(x, \xi, z) = g_3(\xi, \mathcal{T}_{\hat{T}}(x, \xi, z)) = \sqrt{3}(x_2 z_2 + x_5) - (3 + \xi)(x_1 z_1 + x_3 z_2 + x_4) + \sqrt{3}(2 + \xi).$$

This leads to the following standard probust optimization problem:

$$\text{SDC-SPP} : \min_{x \in X} -\pi x_1 x_2 \text{ s.t. } \mathbb{P}\begin{pmatrix} \tilde{g}_1(x, \xi, z) \leq 0 \ \forall z \in B_1(0), \\ \tilde{g}_2(x, \xi, z) \leq 0 \ \forall z \in B_1(0), \\ \tilde{g}_3(x, \xi, z) \leq 0 \ \forall z \in B_1(0) \end{pmatrix} \geq 0.9$$

Please note that we ignored the sign-functions in the definition of $\tilde{g}_i, i = 1, 2, 3$. We are allowed to do that because any feasible decision $x \in X$ with $x_1, x_2 \geq 0$ satisfies $\xi \geq -3$ for all its feasible realizations $\xi \in \Omega(x) := \{\xi \in \Xi \mid \tilde{g}_i(x, \xi, z) \leq 0 \ \forall i = 1, 2, 3, z \in B_1(0)\}$. We can understand that with the following indirect argument:

If there is a feasible decision $x \in X$ with feasible realization $\xi < -3$, then we can distinguish three cases.

Case (i) there exists a feasible realization $\tilde{\xi} > -3$, which implies $D(x) \cap C(\xi) = \emptyset$ or $D(x) \cap C(\tilde{\xi}) = \emptyset$ (see Figure 3.1) what contradicts that ξ and $\tilde{\xi}$ are feasible realizations for $x \in X$.

Case (ii) the realization $\tilde{\xi} = -3$ is feasible. Then the triangle is degenerated and we have to choose $x_1 = x_2 = 0$ leading to $f(x) = 0$.

Case (iii) all feasible realizations satisfy $\xi_3 < -3$. Then we can estimate

$$\varphi(x) \leq \mathbb{P}(\xi_3 \leq -3) = F(-3) \approx 2.87 \cdot 10^{-7}.$$

As we chose $p = 0.9 > 10^{-6}$, x cannot be feasible what is a contradiction.

Consequently, we can ignore the case $\xi < -3$ if we consider feasible decisions $x \in X$.

Analysis (solution existence): Using the representation SDC-SPP, we can use Proposition 1.1.6 to guarantee that SDC-SPP has a solution $x^* \in X$ because all describing functions $\tilde{g}_i$ are continuous w.r.t. $(x, \xi, z) \in X \times \Xi \times \hat{T}$ and therefore satisfy the conditions of the statement. Consequently, $\varphi : X \to [0, 1]$ is upper semi-continuous and the feasible set is closed. As we can w.l.o.g. exchange the unbounded set X by the compact set $\tilde{X} := [0, 4]^2 \times [-2, 2]^3$ and the objective function is continuous, we know by Weierstrass' theorem that there exists an optimal solution $x^* \in X$.

We are allowed to exchange X by $\tilde{X}$ as we chose $p = 0.9 > 0.5$. Because the set of feasible realizations $\Omega(x)$ for any $x \in X$ is described by convex functions it is convex itself and by the symmetry of the given probability distribution $\mathbb{P}(\mu + A) = \mathbb{P}(\mu - A)$ for any $A \in \mathcal{B}$, we have to guarantee that $\mu = 0 \in \Omega(x)$ for any feasible decision $x \in X$. This leads to the deterministic constraint $D(x) \subseteq C(0)$ which implies $y_1 \in [-1, 2]$ and $|y_2| \leq \sqrt{3}$ for all $y \in D(x)$ from a geometrical point of view. These constraints again imply that $x \in \tilde{X}$, where we use that $D(x) = A(x)B_1(0) + b(x) \subseteq C(0)$ and fix e.g. $z_0 = (0, 0) \in B_1(0)$ to get the constraint

$$A(x)z_0 + b(x) = b(x) \in C(0).$$

This fixation of the scenario implies $x_4 \in [-\sqrt{3}, \sqrt{3}] \subseteq [-2, 2]$ and $x_5 \in [-1, 2] \subseteq [-2, 2]$. The fixation of $z_0 \in \{(0,1), (1,0), (-1,0), (0,-1)\}$ leads to the bounds of x_1, x_2 and x_3. Additionally, we know that the feasible set is not empty because the design defined by $x_0 = (0,0,0,0,0)$ lies within all triangles with $\xi \geq -2$ and consequently, we know that $p_{\max} \geq \mathbb{P}(\xi \geq -2) > 0.99 > p = 0.9$. This ensures that x_0 is feasible for our problem instance. All together we know that SDC-SPP has a well-defined solution.

Analysis (convex feasible set): Furthermore, we know by the same arguments as in the last example and by Proposition 1.1.8 that $\varphi : X \to [0,1]$ is a log-concave function. This implies the convexity of the feasible set of SDC-SPP. Altough the objective function is not strictly convex, the following solution step shows that the minimizer of SDC-SPP is unique.

Analysis (solution): As we know that there exists a solution of SDC-SPP, we can try to calculate it. Unfortunately, the standard procedure for solving standard probust optimization problems from Example 1.1.9 is difficult to apply here. One could calculate the worst-case scenarios $z_i^*, i = 1,2,3$, but reformulating the probability evaluation does not lead to a useful expression as far as we experienced.

Consequently, we need a new reformulation of SDC-GPP that allows us to use a new kind of (geometric) arguments.

We change the perspective to find a transformation that does not create a reference set $\hat{T}$ for the design $D(x)$, but that creates a reference set $\tilde{C}$ for the container $C(\xi)$. We choose this reference set $\tilde{C}$ to be the equilateral triangle defined by

$$Q_1 := (-\sqrt{3}, -1), Q_2 := (\sqrt{3}, -1), Q_3 := (0, 2).$$

The corresponding transformation that maps $\tilde{C}$ to $C(\xi)$ for all $\xi \in \Xi$ can be defined as

$$\begin{aligned} \mathcal{T}_{\tilde{C}} : X \times \Xi \times \tilde{C} \to C(\xi), (x, \xi, z) \to \tilde{A}^{-1}(\xi)(z - \tilde{b}(\xi)), \\ \text{where } \tilde{A}(\xi)P_i + \tilde{b}(\xi) = Q_i \text{ for all } i = 1,2,3. \end{aligned}$$

In our example the solution of the linear equality systems yields

$$\tilde{A}(\xi) = \begin{pmatrix} 1 & 0 \\ 0 & \frac{3}{3+\xi} \end{pmatrix}, \tilde{b}(\xi) = \begin{pmatrix} 0 \\ -\frac{\xi}{3+\xi} \end{pmatrix}.$$

The stochastic design-centering problem then looks like

$$\begin{aligned} \tilde{\text{SDC}} : \min_{x \in X} -\pi x_1 x_2 \text{ s.t.} \mathbb{P}\left(\tilde{D}(x,\xi) \subseteq \tilde{C}\right) \geq p, \\ \text{where } \tilde{D}(x,\xi) = \tilde{A}(\xi)D(x) + \tilde{b}(\xi) = \{y \in \mathbb{R}^2 \mid \exists z \in B_1(0) : y = \tilde{A}(\xi)(A(x)z + b(x)) + \tilde{b}(\xi)\}. \end{aligned}$$

From Corollary 4.2 in [22] and Equations (3), (4), (6) from [85], we know that the biggest ellipse within an equilateral triangle is its incircle. Consequently, the optimal choice for fixed $\xi \in \Xi$ is

$$\begin{aligned} A(x^*, \xi) &= \tilde{A}^{-1}(\xi) = \begin{pmatrix} 1 & 0 \\ 0 & 1 + \frac{\xi}{3} \end{pmatrix} \text{ and} \\ b(x^*, \xi) &= -\tilde{A}^{-1}(\xi)\tilde{b}(\xi) = \begin{pmatrix} 0 \\ \frac{\xi}{3} \end{pmatrix}. \end{aligned}$$

This leads to a ξ-depending objective value of $f(x^*,\xi) = -\frac{\pi(3+\xi)}{3}$ which is monotonically decreasing w.r.t. $\xi \in \mathbb{R}$.
Since we have to fulfill the probust constraint with threshold p, we consider exactly the realizations that correspond to the lowest possible objective values. Consequently, we are asking for a lower bound $\underline{\xi} \in \mathbb{R}$ such that:

$$\begin{aligned} &\mathbb{P}([\underline{\xi},\infty)) = p \\ \Leftrightarrow &1 - \mathbb{P}((-\infty,\underline{\xi}]) = p \\ \Leftrightarrow &\underline{\xi} = F^{-1}(1-p) \approx -1.2816 \end{aligned}$$

As we can notice from the original problem SDC: Increasing the height of the triangle by increasing ξ leads to the containment $C(\xi)$ within all triangles $C(\tilde{\xi})$ with $\xi \leq \tilde{\xi}$ as sketched in Figure 5.1. This means that an ellipse within $C(\xi)$ lies within all $C(\tilde{\xi})$ with $\tilde{\xi} \geq \xi$ and consequently the volume maximal ellipse within $C(\underline{\xi})$ is feasible for exactly $100 \cdot p$ percent of the realizations.

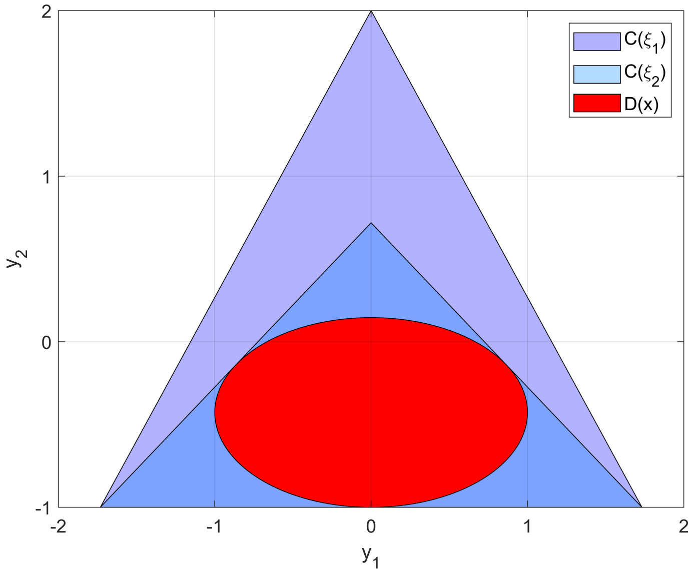


Figure 5.1: Increasing realizations $\xi_1 \geq \xi_2$ ensure containment of design $D(x)$

This means that we can determine the optimal solution of SDC-GPP by

$$A(x^*) = \tilde{A}^{-1}(\underline{\xi}) \text{ and } b(x^*) = -\tilde{A}^{-1}(\underline{\xi})\tilde{b}(\underline{\xi})$$

what implies

$$\begin{aligned} x^* &= \left(1, 1 + \frac{\xi^*}{3}, 0, 0, \frac{\xi^*}{3}\right) \approx (1, 0.5728, 0, 0, -0.4272), \\ f(x^*) &= -\frac{\pi(3+\xi^*)}{3} \approx -1.7996. \end{aligned}$$

As the inner circle of a triangle is defined uniquely by an affine transformation of $B_1(0)$, the affine transformation of $B_1(0)$ to the incircle of $C(\underline{\xi})$ and therefore the solution of this problem is unique as well.

Example: Circle into circle with decision-dependent probability distribution

Problem formulation: Last, but not least we consider the stochastic design-centering problem of putting a circle with uncertain midpoint interpreted as the design

$$D(\xi) = B_r(\xi) = \{y \in \mathbb{R}^2 \mid (y_1 - \xi_1)^2 + (y_2 - \xi_2)^2 \leq r^2\}$$

with a fixed $r \geq 0, \xi \in \mathbb{R}^2$ into a fixed circle interpreted as the container that is described for a fixed $R \geq 0$ by

$$C = B_R(0) = \{y \in \mathbb{R}^2 \mid y_1^2 + y_2^2 \leq R^2\}.$$

The new perspective of this problem is that we assume that the probability distribution varies with our decision $x \in X := \mathbb{R}_{\geq 0}$ via $Z \sim \mathcal{N}((0,0), xI_2)$ and therefore influences the uncertain design $D(\xi)$ indirectly.
As the size of the design is fixed, we are interested in the maximal $x \geq 0$ such that we can guarantee that the design lies within the container with a probability of at least 90%. The stochastic design-centering problem can then be written as

$$\text{SDC} : \max_{x \geq 0} x \text{ s.t. } \mathbb{P}(x)(D(\xi) \subseteq C) \geq p.$$

Problem reformulation (GPP): Before analyzing this problem, we have to reformulate its constraints in a way that we can handle them. Therefore, we rewrite SDC as a generalized probust optimization problem as explained in the beginning of this chapter

$$\text{SDC-GPP} : \min_{x \geq 0} -x \text{ s.t. } \mathbb{P}(x)\left(y_1^2 + y_2^2 - R^2 \leq 0 \;\forall y \in B_r(\xi)\right) \geq 0.9.$$

Problem reformulation (SPP): Next we use the following transformation to the reference probability distribution $\hat{Z} \sim \mathcal{N}((0,0), I_2)$ on $(\hat{\Xi}, \mathcal{B}), \hat{\Xi} = \mathbb{R}^2$ to simplify the probability evaluation

$$\mathcal{T}_{\hat{\Xi}} : X \times \Xi \to \mathbb{R}^2, (x, \omega) \to x\omega.$$

As $\mathcal{T}_{\hat{\Xi}}(x, \cdot)$ is linear for all fixed $x > 0$ w.r.t. $\omega \in \hat{\Xi}$, we can reformulate the inner functions of the original problem as

$$\tilde{\mathbb{P}}\left(y_1^2 + y_2^2 - R^2 \leq 0 \;\forall y \in B_r(x\omega)\right) \geq 0.9.$$

In a second step we use the following transformation to the reference set $\hat{T} = B_1(0)$ to simplify the set-dependency within the probability evaluation

$$\mathcal{T}_{\hat{T}} : X \times \hat{\Xi} \times \hat{T} \to \mathbb{R}^2, (x, \omega, z) \to rz + x\omega.$$

This transformation is surjective for all fixed $x \in X, \xi \in \Omega$ and even a homeomorphism between $\hat{T} = B_1(0)$ and $B_r(x\omega)$.
Therefore, we can consider the following standard probust optimization problem:

$$\begin{aligned}\text{SDC-SPP} : \min_{x \geq 0} -x \text{ s.t. } &\mathbb{P}(g(x, \omega, z) \leq 0 \;\forall z \in B_1(0)) \geq 0.9,\\ \text{where } \tilde{g}(x, \omega, z) &= g(x, \mathcal{T}_{\hat{\Xi}}(x, \omega), \mathcal{T}_{\hat{T}}(x, \mathcal{T}_{\hat{\Xi}}(x, \omega), z))\\ &= (rz_1 + x\omega_1)^2 + (rz_2 + x\omega_2)^2 - R^2\end{aligned}$$

Analysis (solution existence): Using the representation of SDC-SPP, we can use Proposition 1.1.6 to guarantee that this problem has a solution $x^* \in X$ because $\tilde{g}$ is continuous w.r.t. $(x, \omega, z) \in X \times \Omega \times Z$ and therefore satisfy the conditions of the statement. Consequently, $\varphi : X \to [0, 1]$ is upper semi-continuous and the feasible set is closed. As we can w.l.o.g. exchange the unbounded set X by the compact set $\tilde{X} := [0, R]$ and the objective function is continuous, we know by Weierstrass' theorem that there exists an optimal solution $x^* \in \tilde{X} \subseteq X$.
We are allowed to exchange X by $\tilde{X}$ because the probust constraint is continuous and monotonically decreasing w.r.t. $x \geq 0$ with $\lim_{x\to\infty} \varphi(x) = 0$ (see Equation (5.2)). Consequently, there exists a some $\overline{x} \geq 0$ such that $\varphi(x) < p$ for all $x > \overline{x}$, what means that the feasible set of SDC-SPP is compact.
The feasible set is not empty because the decision $x = 0$ leads to the deterministic constraint $B_r(0) \subseteq B_R(0)$ that is fulfilled due to the definition of r and R.
All together, we know that SDC-SPP has a well-defined solution.
Analysis (convex feasible set): Furthermore, we know by the same arguments as in the first example and by Proposition 1.1.8 that $\varphi : X \to [0, 1]$ is a log-concave function. This implies the convexity of the feasible set of SDC-SPP. Because the objective function of SDC-SPP is monotonically decreasing w.r.t. $x \geq 0$, we can calculate the unique minimizer of SDC-SPP in the following solution step.
Analysis (solution calculation): The solution of this problem can be calculated considering the three solution steps of Example 1.1.9:
In a first step we calculate the worst-case scenarios for the inner function. Considering the problem SDC geometrical, it is clear that the worst-case scenario is $z^*(x, \omega) = \frac{\omega}{||\omega||_2}$ (see Figure 5.2) for any $x \in X, \omega \in \hat{\Xi}$.

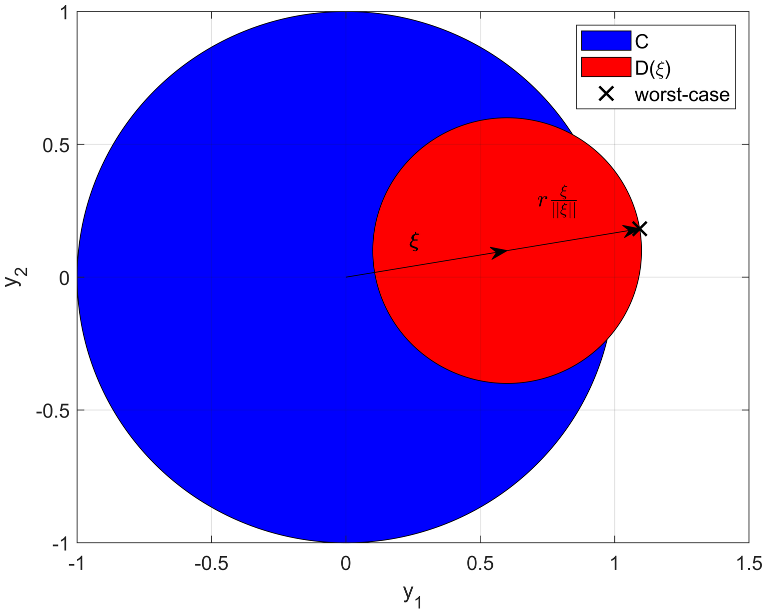


Figure 5.2: Worst-case scenario for a given realization ξ

Consequently, a feasible scenario $\omega \in \hat{\Xi}$ for a fixed decision $x \geq 0$ has to fulfill

$$\begin{aligned}
&\max_{z \in B_1(0)} (rz_1 + x\omega_1)^2 + (rz_2 + x\omega_2)^2 - R^2 \\
&= \left(r\frac{\omega_1}{||\omega||_2} + x\omega_1 \right)^2 + \left(r\frac{\omega_2}{||\omega||_2} + x\omega_2 \right)^2 - R^2 \\
&= \omega_1^2 \left(\frac{r}{||\omega||_2} + x \right)^2 + \omega_2^2 \left(\frac{r}{||\omega||_2} + x \right)^2 - R^2 \\
&= ||\omega||_2^2 \left(\frac{r}{||\omega||_2} + x \right)^2 - R^2 \\
&= (r + ||\omega||_2 x)^2 - R^2 \leq 0 \\
\Leftrightarrow &||\omega||_2 \leq \frac{R-r}{x}.
\end{aligned}$$

This means that we can rewrite the set of feasible realizations as $\Omega(x) = B_{\frac{R-r}{x}}(0)$. Consequently, we can calculate the probability $\varphi(x)$ for $x \geq 0$ switching to polar coordinates:

$$\begin{aligned}
\varphi(x) &= \tilde{\mathbb{P}}((rz_1 + x\omega_1)^2 + (rz_2 + x\omega_2)^2 - R^2 \leq 0 \ \forall z \in B_1(0)) \\
&= \tilde{\mathbb{P}}((r + ||\omega||x)^2 - R^2 \leq 0) \\
&= \int_{\xi \in \Omega(x)} \frac{1}{2\pi} \exp\left(-\frac{\omega_1^2 + \omega_2^2}{2} \right) d\lambda(\xi) \\
&= \int_0^{\frac{R-r}{x}} \int_0^{2\pi} \frac{\tilde{r}}{2\pi} \exp\left(-\frac{\tilde{r}^2}{2} \right) \, d\phi d\tilde{r} \\
&= \int_0^{\frac{R-r}{x}} \tilde{r} \exp\left(-\frac{\tilde{r}^2}{2} \right) \, d\tilde{r} \\
&= \left[-\exp\left(-\frac{\tilde{r}^2}{2} \right) \right]_0^{\frac{R-r}{x}} \\
&= 1 - \exp\left(-\frac{(R-r)^2}{2x^2} \right) \qquad (5.2)
\end{aligned}$$

Since φ is monotonically decreasing w.r.t. $x \geq 0$, the (unique) solution x^* of this problem can be calculated by

$$\varphi(x^*) = 1 - \exp\left(-\frac{(R-r)^2}{2(x^*)^2} \right) = p$$

what can be reformulated as

$$\Leftrightarrow x^* = \sqrt{-\frac{(R-r)^2}{2\ln(1-p)}}.$$

In Section 5.3 we are interested in the problem instance defined by $R = 15, r = 3$ and $p = 0.9$ which leads to the minimizer

$$x^* \approx 5.592.$$

5.2 Modified subset schemes

As we have considered three examples of stochastic design-centering problems in the last section and solved them analytically, we are now interested in solving them numerically to compare the performance of the corresponding solution schemes. We modify the lower-level adaptive discretization approach (LLAD) and the high-level adaptive discretization approach (HLAD) introduced in Section 2.3 to make the solution process more efficient. We are mainly inspired by the question:

> Can we use the maximium structure of the inner function $g = \max_{i\in I} g_i$?

We start with a variant of the LLAD (see Equation (2.5)). We specify the subset scheme $\Phi = (\Phi_k)_{k\in\mathbb{N}}$ in the probust discretization algorithm introduced in Section 2.2 as

$$\Phi_k : X \times 2^T \to 2^T, (x, S) \mapsto S \cup \bigcup_{i\in I} S^*_{k,i} \tag{5.3}$$

where $S^*_{k,i} := \{t^*_{k,i}(\xi) \mid \xi \in \dot{\Xi}_{k,i}\}$, $t^*_{k,i}(\xi) \in \arg\max_{t\in T} g_i(x, \xi, t)$ are sets of worst-case scenarios, depending on the realizations $\xi \in \dot{\Xi}_{k,i}$, where $\emptyset \neq \dot{\Xi}_{k,i} \subseteq \Xi$ for each $k \in \mathbb{N}, i \in I$. By definition this scheme is an increasing subset scheme.
If $|\dot{\Xi}_{k,i}| < \infty$ for all $k \in \mathbb{N}, i \in I$, this scheme is a discretization scheme.

Lemma 5.2.1 (Convergence of maximum-structure using variant of LLAD) *Choosing Φ as the maximum-structure using variant of LLAD, where $g_i : X \times \Xi \times T \to \mathbb{R}$ are continuous functions w.r.t. $(x, \xi, t) \in X \times \Xi \times T$ fulfilling Assumption 2.1.6 for all $x \in X, i \in I$ and $\emptyset \neq \dot{\Xi}_{k,i}$ for all $k \in \mathbb{N}, i \in I$ is defined by realizations that are randomly i.i.d. chosen according to the random variable Z given in the probust optimization problem, any accumulation point $\overline{x}$ of $(x_k)_{k\in\mathbb{N}}$ is $\mathbb{P}$-almost surely an optimal solution of the original probust optimization problem.*

Proof. We prove this claim by using Theorem 2.2.3:
We show that the candidate-condition 2.1.9 is fulfilled $\mathbb{P}$-almost surely for any point $x \in X$ and therefore especially for any accumulation point $\overline{x} \in X$ of $(x_k)_{k\in\mathbb{N}}$.
Please note that $t^*_{k,i}(\xi) \in \arg\max_{t\in T} g_i(x_k, \xi, t)$ is well-defined for all $\xi \in \Xi, k \in \mathbb{N}, i \in I$ as T was assumed to be compact and g was assumed to be continuous w.r.t. $t \in T$ for all $(x, \xi) \in X \times \Xi$. Furthermore, because $X \subseteq \mathbb{R}^n$ is compact, the sequence $(x_k)_{k\in\mathbb{N}}$ has at least one accumulation point.
Now we fix an arbitrary $x \in X$ and argue indirectly by assuming

$$\exists t_f \in T, \epsilon > 0 : \varphi(x, S) - \varphi(x, S \cup \{t_f\}) \geq \epsilon.$$

Considering the set

$$\Omega = \{\xi \in \Xi \mid g(x, \xi, s) \leq 0 \text{ for all } s \in S \wedge g(x, \xi, t_f) > 0\}$$

we know by the continuity of g w.r.t. $(x, \xi, t) \in X \times \Xi \times T$ that $\Omega \in \mathcal{A}$ is measurable and by Theorem 2.1.10 we know that

$$\mathbb{P}(\Omega) = \varphi(x, S) - \varphi(x, S \cup \{t_f\}) \geq \epsilon.$$

Additionally, we know by Assumption 2.1.6 and the continuity of g w.r.t. $t \in T$ that there exists a set $\tilde{\Omega} \in \mathcal{A}$ such that

$$\begin{aligned}
&\tilde{\Omega} = \{\xi \in \Xi \mid g\left(x, \xi, s\right) < 0 \text{ for all } s \in S \wedge g\left(x, \xi, t_f\right) > 0\} \text{ and}\\
&\mathbb{P}\left(\tilde{\Omega}\right) = \mathbb{P}\left(\Omega\right) \geq \epsilon.
\end{aligned}$$

Because $g\left(x, \cdot, t_f\right)$ is continuous w.r.t. $\xi \in \Xi$ and $\tilde{\Omega} \neq \emptyset$ due to $\epsilon > 0$, we can find a $\xi_f \in \tilde{\Omega}$ and a $r > 0$ such that

$$\begin{aligned}
g\left(x, \xi, t_f\right) > 0 \text{ for all } \xi \in B_r\left(\xi_f\right) \text{ and}\\
\max_{s \in \mathrm{cl}(S)} g\left(x, \xi, s\right) < 0 \text{ for all } \xi \in B_r\left(\xi_f\right).
\end{aligned}$$

Since $\xi \in \mathit{supp}\left(\mathbb{P}\right) = \{\xi \in \Xi \mid \forall r > 0 : \mathbb{P}(B_r(\xi)) > 0\}$, we can introduce

$$\varepsilon := \mathbb{P}\left(B_r\left(\xi_f\right)\right) > 0.$$

As we choose at least one realization in each iteration $k \in \mathbb{N}$ and the choice of new scenarios $\xi_k \in \Xi$ is i.i.d. the probability to choose a realization $\xi_k \in B_r\left(\xi_f\right)$ for any $k \in \mathbb{N}$ is

$$\begin{aligned}
\mathbb{P}\left(\exists k \in \mathbb{N} : \xi_k \in B_r\left(\xi_f\right)\right) &= 1 - \mathbb{P}\left(\forall k \in \mathbb{N} : \xi_k \notin B_r\left(\xi_f\right)\right)\\
&= \lim_{k \to \infty} 1 - \left(1 - \epsilon\right)^k\\
&= 1.
\end{aligned}$$

Therefore, we $\mathbb{P}$-almost surely pick a realization $\omega \in B_r\left(\xi_f\right)$ at some iteration $k \in \mathbb{N}$. Assuming we pick this realization in iteration $K \in \mathbb{N}$, we know that we add

$$t^*_{K,i}\left(\omega\right) := \arg\max_{t \in T} g_i\left(x, \omega, t\right) \in T_{K+1,i} \subseteq T_{K+1} \subseteq \bigcup_{k=1}^{\infty} \bigcup_{i \in I} T_{k,i} = S$$

to our discretization and because of

$$\begin{aligned}
g\left(x, \omega, t_f\right) &\leq \max_{t \in T} g\left(x, \omega, t\right)\\
&= \max_{t \in T} \max_{i \in I} g_i\left(x, \omega, t\right)\\
&= \max_{i \in I} \max_{t \in T} g_i\left(x, \omega, t\right)\\
&= \max_{i \in I} g_i\left(x, \omega, t^*_{K,i}\left(\omega\right)\right)\\
&= \max_{i \in I} \max_{t \in T^*_K} g_i\left(x, \omega, t\right)\\
&= \max_{t \in T^*_K} \max_{i \in I} g_i\left(x, \omega, t\right)\\
&= \max_{t \in T^*_K} g\left(x, \omega, t\right)
\end{aligned}$$

we know that (ω, t_f) has to fulfill

$$\begin{aligned}&g(x,\omega,t_f) > 0 \text{ because } \omega \in B_r(\xi_f) \text{ and}\\&g(x,\omega,t_f) \leq \max_{t\in T_K^*} g(x,\omega,t) \leq \max_{t\in \mathrm{cl}(S)} g(x,\omega,t) < 0.\end{aligned}$$

This is a contradiction.
Therefore, our basic assumption is $\mathbb{P}$-almost surely wrong and consequently the candidate-condition holds for any point $x \in X$ $\mathbb{P}$-almost surely, especially for all accumulation points $\overline{x} \in X$ of $(x_k)_{k\in\mathbb{N}}$. Using Theorem 2.2.3 the claim is true. □

This proof shows that we can use the LLAD on each function $g_i, i \in I$ separately and still guarantee the convergence of the scheme.
It is just noted here that we produce at least one discretization point $t_{k,i}$ per index $i \in I$ and iteration $k \in \mathbb{N}$ with this subset scheme. To reduce the number of discretization points $|T_k|$, we can add the following *importance-sampling-condition*:

$$t_{k,i}^*(\xi) \in S_{k,i}^* \Leftrightarrow g_i\left(x_k, \xi, t_{k,i}^*(\xi)\right) > 0$$

Now we consider a similar approach with the HLAD (see Equation (2.6)). Therefore, we specify the subset scheme $\Phi = (\Phi_k)_{k\in\mathbb{N}}$ in the probust discretization algorithm introduced in Section 2.2 as

$$\Phi_k : X \times 2^T \to 2^T, (x, S) \mapsto S \cup \bigcup_{i\in I} S_{k,i}^*, \tag{5.4}$$

where $S_{k,i}^* := \{t_1^*, ..., t_{n_{k,i}}^*\}$, $\left(t_1^*, ..., t_{n_{k,i}}^*\right) \in \arg\min_{\left(t_1,...,t_{n_{k,i}}\right)\in T^{n_{k,i}}} \varphi_i\left(x, T_{k,i} \cup \{t_1, ..., t_{n_{k,i}}\}\right)$ are the worst-case scenarios considering the probability of $x \in X$ given the subsets

$$T_{k,i} = T_0 \cup \bigcup_{j=1}^{k-1} S_{j,i}^* \subseteq S \subseteq T$$

for each $k \in \mathbb{N}, i \in I$ of the functions

$$\varphi_i(x, S) := \mathbb{P}\left(g_i(x, \xi, t) \leq 0 \ \forall t \in S\right)$$

and $(n_{k,i})_{k\in\mathbb{N}}$ are sequences of natural numbers for all $i \in I$.
By definition this scheme is an increasing subset scheme. If $|T_0| < \infty$ for all $k \in \mathbb{N}$, this scheme is a discretization scheme.

Lemma 5.2.2 (Convergence of maximum-structure using variant of HLAD)
Choosing Φ as the maximum-structure using variant of HLAD, where $g_i : X \times \Xi \times T \to \mathbb{R}$ are continuous functions w.r.t. $(x, \xi, t) \in X \times \Xi \times T$ fulfilling Assumption 2.1.6 for all $x \in X, i \in I$, any accumulation point $\overline{x}$ of $(x_k)_{k\in\mathbb{N}}$ is an optimal solution of the original probust optimization problem.

Proof. We prove this claim by using Theorem 2.2.3:
We show that the candidate-condition is fulfilled for any accumulation point $\overline{x} \in X$ of $(x_k)_{k\in\mathbb{N}}$ indirectly.
Please note that with Assumption 2.1.6 and by the compactness of T the discretization scheme is well-defined for all $k \in \mathbb{N}$. Furthermore due to the compactness of $X \subseteq \mathbb{R}^n$, the sequence $(x_k)_{k\in\mathbb{N}}$ has at least one accumulation point. Without loss of generality we assume that the whole sequence $(x_k)_{k\in\mathbb{N}}$ has just one accumulation point. Otherwise we switch notations to consider a fixed converging subsequence.

We fix the accumulation point $\overline{x} \in X$, set $S := \lim_{k\to\infty} T_k$ and assume

$$\exists t_f \in T, \epsilon > 0 : \varphi(\overline{x}, S) - \varphi(\overline{x}, S \cup \{t_f\}) \geq \epsilon. \tag{5.5}$$

In the next step we show that this implies that

$$\exists j \in I : \varphi_j(\overline{x}, S) - \varphi_j(\overline{x}, S \cup \{t_f\}) \geq \frac{\epsilon}{|I|}. \tag{5.6}$$

Therefore, we use that for any index set I and arbitrary families of measurable sets $(A_i)_{i\in I}, (B_i)_{i\in I}$ it holds

$$\left(\bigcap_{i\in I} A_i\right) \setminus \left(\bigcap_{i\in I} B_i\right) \subseteq \bigcup_{i\in I} (A_i \setminus B_i). \tag{5.7}$$

We show statement (5.7) by fixing any $x \in \left(\bigcap_{i\in I} A_i\right) / \left(\bigcap_{i\in I} B_i\right)$. By definition this means that $x \in A_i$ for all $i \in I$ and there exists some $k \in I$ such that $x \notin B_k$. Consequently, $x \in A_k \setminus B_k$ and $x \in \bigcup_{i\in I} (A_i \setminus B_i)$. As $x \in \left(\bigcap_{i\in I} A_i\right) \setminus \left(\bigcap_{i\in I} B_i\right)$ was chosen arbitrarily, the statement is proven.

Using Statement (5.7) and the notation $\Omega_i(\overline{x}, S) = \{\xi \in \Xi \mid g_i(\overline{x}, \xi, t) \leq 0 \text{ for all } t \in S\}$, we can estimate:

$$\begin{aligned}
0 < \epsilon &\leq \varphi(\overline{x}, S) - \varphi(\overline{x}, S \cup \{t_f\}) \\
&= \mathbb{P}\left(\Omega(\overline{x}, S) \setminus \Omega(\overline{x}, S \cup \{t_f\})\right) \\
&= \mathbb{P}\left(\bigcap_{i\in I} \Omega(\overline{x}, S) \setminus \bigcap_{i\in I} \Omega(\overline{x}, S \cup \{t_f\})\right) \\
&\leq \mathbb{P}\left(\bigcup_{i\in I} \left(\Omega_i(\overline{x}, S) \setminus \Omega_i(\overline{x}, S \cup \{t_f\})\right)\right) \\
&\leq \sum_{i\in I} \mathbb{P}\left(\Omega_i(\overline{x}, S) \setminus \Omega_i(\overline{x}, S \cup \{t_f\})\right) \\
&= \sum_{i\in I} \varphi_i(\overline{x}, \Omega(S)) - \varphi_i(\overline{x}, \Omega(S \cup \{t_f\})) \\
&\leq |I| \cdot \max_{i\in I}\{\varphi_i(\overline{x}, \Omega(S)) - \varphi_i(\overline{x}, \Omega(S \cup \{t_f\}))\}
\end{aligned}$$

If Statement (5.6) would be wrong, the last estimation would lead to $\epsilon < \epsilon$ and therefore would be a contradiction w.r.t. $\epsilon > 0$. Consequently Statement (5.6) is true.

In a last step we show that there does not exist any $j \in I$ such that Statement (5.6) holds. As Assumption (5.5) implies this statement, we contradict the main assumption and prove the claim.
We use Theorem 2.1.11 and Proposition 2.1.18 to estimate for an arbitrary $i \in I$:

$$\begin{aligned}
&\lim_{k\to\infty} \varphi_i\left(\overline{x}, T_{k,i}\right) - \varphi_i\left(\overline{x}, T_{k,i} \cup \{t_f\}\right) \\
=&\lim_{k\to\infty} \varphi_i\left(\overline{x}, T_{k,i}\right) - \varphi_i\left(x_k, T_{k,i} \cup \{t_f\}\right) + \underbrace{\varphi_i\left(x_k, T_{k,i} \cup \{t_f\}\right) - \varphi_i\left(\overline{x}, T_{k,i} \cup \{t_f\}\right)}_{\to 0 \text{ due to Proposition 2.1.14}} \\
=&\lim_{k\to\infty} \varphi_i\left(\overline{x}, T_{k,i}\right) - \varphi_i\left(x_k, T_{k,i} \cup \{t_f\}\right) \\
=&\lim_{k\to\infty} \varphi_i\left(\overline{x}, T_{k,i}\right) - \varphi_i\left(x_k, T_{k+1,i}\right) + \underbrace{\varphi_i\left(x_k, T_{k+1,i}\right) - \varphi_i\left(x_k, T_{k,i} \cup \{t_f\}\right)}_{\leq 0 \text{ due to the definition of } T_{k+1,i}, t^*_{k,i}} \\
\leq&\lim_{k\to\infty} \varphi_i\left(\overline{x}, T_{k,i}\right) - \varphi_i\left(x_k, T_{k+1,i}\right) \\
=&\lim_{k\to\infty} \underbrace{\varphi_i\left(\overline{x}, T_{k,i}\right) - \varphi_i\left(\overline{x}, T_{k+1,i}\right)}_{\to 0 \text{ due to Theorem 2.1.11}} + \underbrace{\varphi_i\left(\overline{x}, T_{k+1,i}\right) - \varphi_i\left(x_k, T_{k+1,i}\right)}_{\to 0 \text{ due to Proposition 2.1.14}} \\
=&0
\end{aligned}$$

By Proposition 2.1.18 and the candidate-condition this implies that $S_i = \lim_{k\to\infty} T_{k,i}$ satisfies

$$\varphi_i\left(\overline{x}, T\right) = \varphi_i\left(\overline{x}, S_i\right).$$

As $S_i \subseteq S$ for all $i \in I$, we also know that $\varphi_i\left(\overline{x}, S\right) = \varphi_i\left(\overline{x}, T\right) = \varphi_i\left(\overline{x}, S \cup \{t_f\}\right)$ and consequently statement (5.6) cannot be fulfilled by any $\epsilon > 0$. This means that the assumption (5.5) is wrong and therefore the claim holds. □

In the next section, we use the introduced schemes and compare them to each other in terms of precision of the iterates, number of discretization points and running time.

5.3 Numerical results of probust solution methods

We now solve the stochastic design-centering problems introduced in Section 5.1 and compare the numerical results with the analytical solutions. We comment on remarkable numerical results directly, while we focus on problem overlapping solution behavior in the end of this section.

On the one hand, we are interested in the solution precision of the outer and inner iterates $\underline{x}^*, \overline{x}^* \in X$. On the other hand, we take care of the chosen discretization points $T_k \subseteq \hat{T}$ and the time needed to solve different subproblems of the algorithms. Here, we denote the time to calculate new discretization points $T_k^* \subseteq S$ by t_S, the time to find the current outer iterate $\underline{x}_k$ by $\underline{t}$ and the time to find the current inner iterate $\overline{x}_k$ by $\overline{t}$ for $k \in \mathbb{N}$. We are not especially interested in the sets of feasible realizations that are implied by the subsets $T_k, k \in \mathbb{N}$ because they are already characterized in the last section.
We compare the following algorithms with each other:

1. Sandwiching with the uniform discretization scheme (see Equation (2.4)) starting with three discretization points per dimension.

2. Sandwiching with the maximum-structure using variant of LLAD (see Equation (5.3)) with ten discretization points per constraint per iteration.

3. Sandwiching with the maximum-structure using variant of HLAD (see Equation (5.4)) with one discretization point per constraint per iteration.

4. Special set-approximation algorithms considering the structure of the sets of feasible realizations that were discussed in the Section 5.1.

We stop the discretization schemes, if the relative change of the minimizer w.r.t. $||.||_2$-norm is below the precision of $\epsilon = 10^{-6}$ or a maximal solving time from 60 minutes is exceeded.
We start all discretization schemes with an initial discretization of $T_0 = \emptyset$ if not stated differently and with a feasible initial point x_0 that is defined according to the single problems.
We work with a self-written solver to solve these problems. This self-written solver uses MATLAB's optimization toolbox and the method *fmincon* to solve the nonlinear optimization problems which appear as subproblems in the individual solving procedures. Additionally, we used the spherical-radial decomposition (SRD) introduced in Section 1.1 to evaluate the probability constraints in connection with probust subset schemes. For one dimensional normal distributed random vectors, we can calculate the probabilities (and their gradients) exactly by the two samples $\{\pm 1\} = \mathbb{S}^0$. For the two dimensional case, we consider 100 equally distributed points of the unit circle $\mathbb{S}^1$ defined by the vectors $v = (cos(2\pi\alpha_i), sin(2\pi\alpha_i)), \alpha_i = 0, 0.01, ..., 0.99$. With this sample we use the approximation given by (1.3).

Example: Circle in uncertain triangle

We have already seen that the unique analytical solution of this problem is

$$\begin{aligned} x^* &\approx (0.1417, 0, 0.4707) \text{ and} \\ f(x^*) &\approx -6.308 \cdot 10^{-2}. \end{aligned}$$

We start all discretization schemes with $x_0 := (0.05, 0, 05, 0.4)$ and $T_0 := \emptyset$.
Using the sandwiching algorithm 7 with the uniform discretization approach defined by $d_1 = 1$ and $d_{k+1} = \frac{d_k}{2}$ for all $k \in \mathbb{N}$, we get the results listed in Table 5.1 and in Table 5.2. While the first table highlights the number of discretization points with the corresponding inner and outer approximations, the second table focuses on the running times.

# points in $[-1,1]^2$	# points in $B_1(0)$	$\underline{x}^*$	$\overline{x}^*$
$(1+2^1)^2 = 9$	5	$(0.1710, 0, 0.5)$	$(0.1417, 0, 0.4707)$
$(1+2^2)^2 = 25$	13	$(0.1710, 0, 0.5)$	$(0.1417, 0, 0.4707)$
$(1+2^3)^2 = 81$	49	$(0.1520, 0, 0.4810)$	$(0.1417, 0, 0.4707)$
$(1+2^4)^2 = 289$	197	$(0.1440, 0, 0.4730)$	$(0.1417, 0, 0.4707)$

Table 5.1: Iterates generated by a sandwiching approach with a uniform discretization scheme

# points in $[-1,1]^2$	# points in $B_1(0)$	$\underline{t}$ [in sec]	$\bar{t}$ [in sec]	t_{total} [in sec]
$(1+2^1)^2 = 9$	5	220	3.95	224
$(1+2^2)^2 = 25$	13	587	4.72	592
$(1+2^3)^2 = 81$	49	2210	15.2	2230
$(1+2^4)^2 = 289$	197	9420	43.1	9460

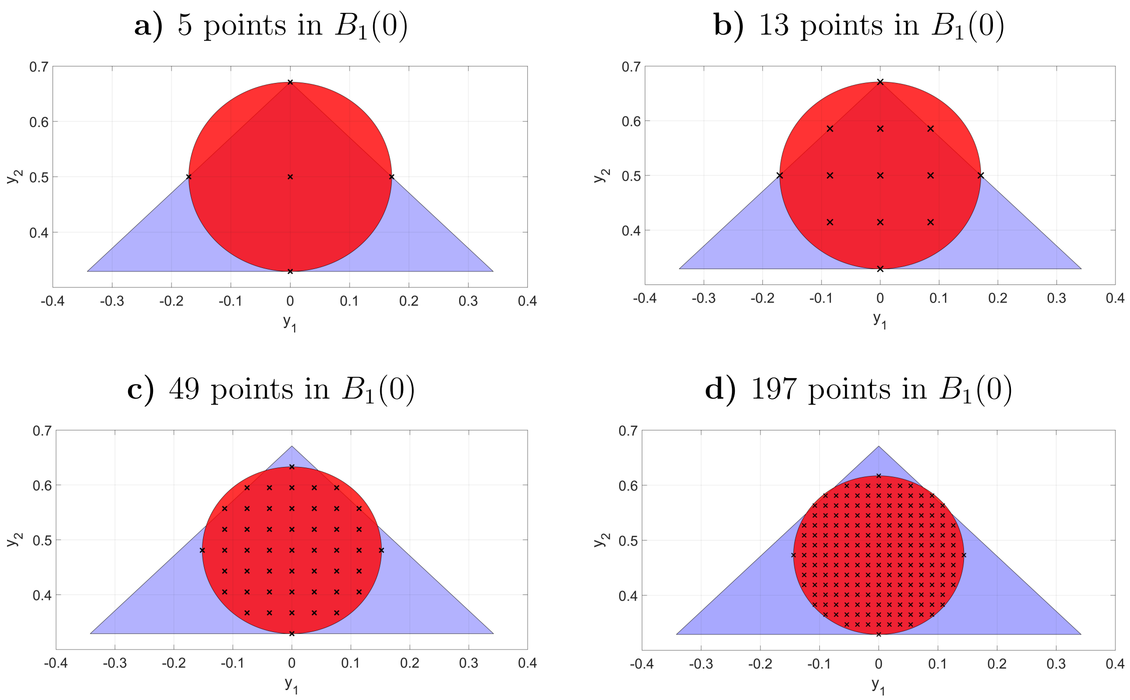
Table 5.2: Computation times using a sandwiching approach with a uniform discretization scheme

Considering the scenario space T, we can visualize the (discretized) design-centering condition for different iterations of the uniform discretization scheme by Figure 5.3. In this figure, the discretized design (black crosses) as a subset of the original design (red circle) should lie within a reference container (blue triangle). The specific definition of the reference container

$$\overline{C} = C(\xi^*) \cap C(-\xi^*),$$

with $\xi^* \approx 0.3290$ is inspired by the analysis of this problem in Section 5.1.

Next, we are interested in the results of the maximum-structure using variants of LLAD and HLAD. Because the maximizers of $\tilde{g}_i(x, \xi, \cdot), i = 1, 2, 3$ are independent of the realization $\xi \in \mathbb{R}$, we know that both approaches lead to the same worst-case scenarios. Because we know that these worst-case scenarios are also independent w.r.t. $x \in X$, the approaches find the optimal solution in their first iteration (see Table 5.3 - Table 5.5).



a) 5 points in $B_1(0)$

b) 13 points in $B_1(0)$

c) 49 points in $B_1(0)$

d) 197 points in $B_1(0)$

Figure 5.3: The uniform discretization (black crosses) of the corresponding design $D(\underline{x}_k)$ (red set) lies within the container $\overline{C}$ (blue set), but $D(\underline{x}_k) \cap \overline{C} \neq D(\underline{x}_k)$

T_k^*	$\underline{x}^*$	$\overline{x}^*$
$(0,-1), \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right), \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$	$(0.1417, 0, 0.4707)$	$(0.1417, 0, 0.4707)$

Table 5.3: Iterates generated by a sandwiching approach with the maximum structure using LLAD/HLAD variant

T_k^*	t_{S} [in sec]	$\underline{t}$ [in sec]	$\bar{t}$ [in sec]	t_{total} [in sec]
$(0,-1), \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right), \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$	1.26	137	3.23	141

Table 5.4: Computation times using a sandwiching approach with the maximum structure using LLAD variant

T_k^*	t_{S} [in sec]	$\underline{t}$ [in sec]	$\bar{t}$ [in sec]	t_{total} [in sec]
$(0,-1), \left(-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right), \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$	14.5	149	3.01	166

Table 5.5: Computation times using a sandwiching approach with the maximum structure using HLAD variant

These solution schemes therefore calculated the (finite) number of worst-case scenarios. The geometric interpretation of this result is seen in Figure 5.4.

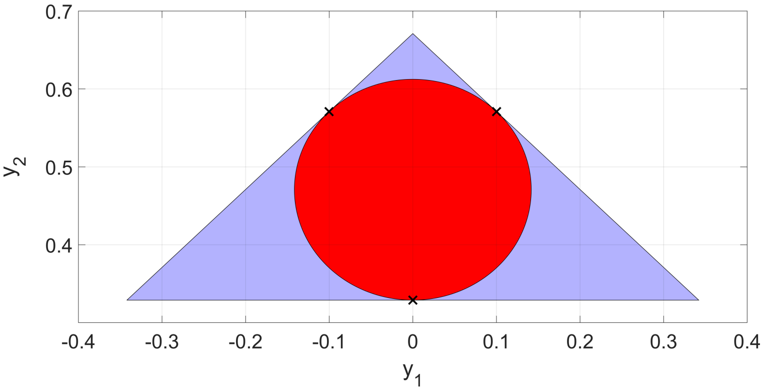


Figure 5.4: The adaptive discretizations (black crosses) and the corresponding design $D(\underline{x}_k)$ (red set) lie within the container $\overline{C}$ (blue set)

It is noticeable that all discretization approaches imply the same set of feasible realizations $\Omega = [-\xi^*, \xi^*]$ with $\xi^* = \sigma F^{-1}\left(\frac{1+p}{2}\right) \approx 0.3290$ which is also the set of feasible realizations of the analytical solution $\Omega(x^*)$. . Consequently, the inner set-approximation problems solved in each iteration can search directly for the solution of the original SDC. This example is a special case because the optimization variables $x_2, x_3 \in \mathbb{R}$ do not effect the objective value and is therefore just used to ensure the feasibility of the decision part $x_1 \geq 0$. As there is one "optimal" set of feasible realizations $\Omega \approx [-0.3290, 0.3290]$ for all $x_1 \geq 0$, the choice of $x_2, x_3 \in \mathbb{R}$ guarantees this form of the set of feasible realizations independent of the choice of $x_1 \geq 0$.

For the set-approximation approach, we use the knowledge we gained by the analysis of the solution in Section 5.1 and define

$$\Delta := \{\delta \in \mathbb{R}^2 \mid -5 \leq \delta_1 \leq \delta_2 \leq 5\},$$
$$D : X \times \Delta \to \mathbb{R}, (x, \delta) \mapsto [\delta_1, \delta_2].$$

Because $\tilde{g}_1$ is monotonically increasing w.r.t. $\xi \in \mathbb{R}$, while $\tilde{g}_i, i = 2, 3$ are monotonically decreasing w.r.t. $\xi \in \mathbb{R}$, we know that the set of feasible realizations is an interval due to Proposition 3.1.7 and

$$\Omega(x) = \Omega(x, T) = \bigcap_{i=1}^{3} \Omega_i(x, T) = (-\infty, \overline{\xi}_1(x)] \cap [\underline{\xi}_2(x), \infty) \cap [\underline{\xi}_3(x), \infty)$$

for some decision dependent bounds $\overline{\xi}_1, \underline{\xi}_2, \underline{\xi}_3 : X \to \mathbb{R}$. Therefore, we can reformulate the probability evaluation as

$$\mathbb{P}(D(x, \delta)) = \mathbb{P}([\delta_1, \delta_2]) = F(5\delta_2) - F(5\delta_1).$$

The result are noted in Table 5.6 and will be commented on in the end of this section.

Solution x^*	design variable δ^*	t_{total} in [sec]
$(0.1417, 0, 0.4707)$	$(-0.3290, 0.3290)$	8.78

Table 5.6: Numerical results of set-approximation approach for SDC circle in uncertain triangle

Example: Ellipse in uncertain triangle

We have already seen that the unique analytical solution of this problem is

$$\begin{aligned} x^* &\approx (1, 0.5728, 0, 0, -0.4272) \text{ and} \\ f(x^*) &\approx -1.800. \end{aligned}$$

To reduce computation times and make the problem more stable, we fix $x_3^* = x_4^* = 0$ in the optimization process and add the constraints $x_1 \in [-1, 1], x_2 \in [-1, 1], x_5 \in [-1, 2]$ and $x_5 - x_2 \geq -1$.
We start all discretization schemes with $x_0 := (0.25, 0.25, 0, 0, 0)$ and $T_0 := \emptyset$.
Using the sandwiching based on the uniform discretization approach, we list the iterates over the iterations in Table 5.7 and the corresponding computation times in Table 5.8.

# points in		Approximations	
$[-1,1]^2$	$B_1(0)$	$\underline{x}^*$	$\overline{x}^*$
$(1+2^1)^2 = 9$	5	$(0.8660, 0.8592, -0.1408)$	$(1, 0.5728, -0.4272)$
$(1+2^2)^2 = 25$	13	$(0.8660, 0.8592, -0.1408)$	$(1, 0.5728, -0.4272)$
$(1+2^3)^2 = 81$	49	$(1.000, 0.6496, -0.3504)$	$(1, 0.5728, -0.4272)$
$(1+2^4)^2 = 289$	197	$(1.000, 0.5996, -0.4004)$	$(1, 0.5728, -0.4272)$

Table 5.7: Iterates generated by a sandwiching approach with a uniform discretization scheme

# points in		time [in sec]		
$[-1,1]^2$	$B_1(0)$	$\underline{t}$	$\overline{t}$	t_{total}
$(1+2^1)^2 = 9$	5	78.2	10.7	89.0
$(1+2^2)^2 = 25$	13	207	14.3	221
$(1+2^3)^2 = 81$	49	260	33.3	293
$(1+2^4)^2 = 289$	197	1460	105	1570

Table 5.8: Computation times using a sandwiching approach with a uniform discretization scheme

Next to the uniform discretization approach, we are interested in the results of the maximum-structure using variants of LLAD and HLAD. As in this example the worst-case scenarios $t^* = t^*(x, \xi)$ depend on the decision x and also on the realization of ξ, we cannot expect the LLAD to find the same points as HLAD. Furthermore, we can only expect to find approximations of the minimizer of the original problem in contrary to the last example.
The iterates and computation times for the sandwiching based on the maximum structure using LLAD variant can be found in Table 5.9 and 5.10 respectively.

Iteration k	$\underline{x}^*$	$\overline{x}^*$
1	$(0.9106, 0.6564, -0.3436)$	$(1, 0.5728, -0.4272)$
2	$(0.9938, 0.5765, -0.4235)$	$(1, 0.5728, -0.4272)$
3	$(0.9938, 0.5765, -0.4235)$	$(1, 0.5728, -0.4272)$
4	$(1, 0.5728, -0.4272)$	$(1, 0.5728, -0.4272)$

Table 5.9: Iterates generated by a sandwiching approach with the maximum using variant of LLAD

Iteration k	t_S [in sec]	$\underline{t}$ [in sec]	$\bar{t}$ [in sec]	t_{total} [in sec]
1	0.892	1450	23.8	1480
2	0.931	1720	6.64	1730
3	1.01	498	20.4	519
4	0.910	655	23.9	680

Table 5.10: Computation times using a sandwiching approach with the maximum structure using LLAD variant

Because we generate 10 random picks in each iteration per constraint, we create up to 30 worst-case scenarios in each iteration in the LLAD approach. How these discretization points are chosen (in one run of random picks) can be seen in Figure 5.5.

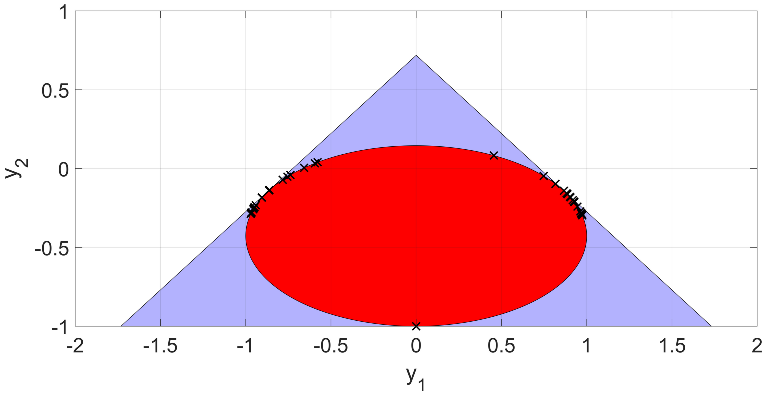


Figure 5.5: LLAD discretization (black crosses) after two iterations with corresponding design $D(x_k)$ (red set) lies within the container $C(\xi^*), \xi^* = 2 - F^{-1}(0.9)$ (blue set)

We see that the discretization points induced by the inner functions g_2, g_3 are changing, but stay in the same area although the triangle moves for different realizations $\xi \in \mathbb{R}$. Since the inner g_1 is independent of $\xi \in \mathbb{R}$, the corresponding worst-case scenario is fixed as in the last example.

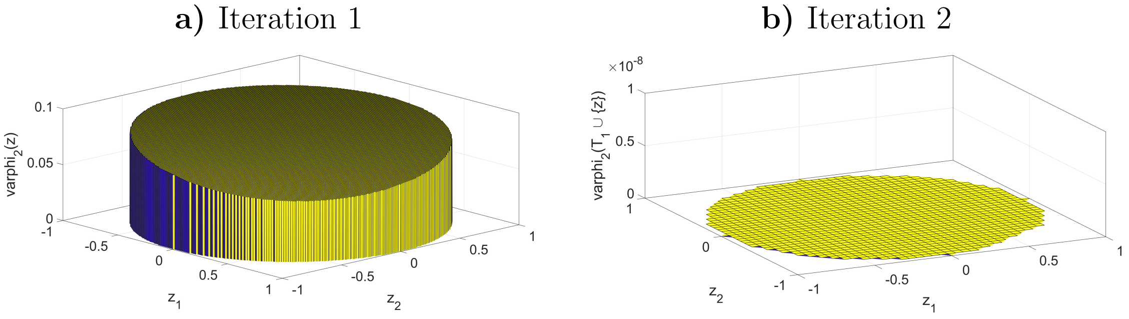
The results for the sandwiching approach based on the maximum structure using HLAD variant are considered in Table 5.11.

Iteration	Approximations		time [in sec]			
k	$\underline{x}^*$	$\overline{x}^*$	t_S	$\underline{t}$	$\overline{t}$	t_{total}
1	$(1, 0.6227, -0.3773)$	$(1, 0.5728, -0.4272)$	11.7	20.2	9.11	41.0

Table 5.11: Iterates and computation times using a sandwiching approach with the maximum structure using HLAD variant

This table emphasizes that the solver does not find new discretization points starting from $t_0 = (0, 0)$ in the second iteration. Although the discretization points in T_0^* might be chosen optimal for the decision x_0, we have seen in Section 5.1 that the worst-case scenarios depend on the decision. Therefore, we expect to find new points in the second iteration.

That we cannot find these points numerically, can be explained by visualizing the optimization problem of choosing a new discretization point in the second iteration (see Figure 5.6).


a) Iteration 1

b) Iteration 2

Figure 5.6: Effect on $\varphi_2(\underline{x}_k, T_k)$ of adding a discretization point to the current discretization set using the HLAD variant

Here the function $\varphi_2(\underline{x}_1, T_1 \cup \{\cdot\})$ is close to zero and nearly constant, especially around the initial point $t_0 = (0, 0)$. Consequently, a gradient based optimizer does not find the minimizer and returns the initial point t_0.

Again all discretization approaches imply the design $\Omega(x_k, T_k) = [\xi^*, \infty)$ with lower bound $\xi^* = F^{-1}(1-p)$ which is also the set of feasible realizations of the analytical solution such that the set-approximation problems find the minimizer of the original problem directly.

For the set-approximation algorithm we use the knowledge we gained by the analysis of the solution in the last section and define

$$\Delta := \{\delta \in \mathbb{R} \mid -5 \leq \delta \leq 5\} \text{ and } D : X \times \Delta \to \mathbb{R}, (x, \delta) \mapsto [\delta, \infty).$$

Because the triangle $C(\xi_2)$ contains the triangle $C(\xi_1)$ for $\xi_1, \xi_2 \in \mathbb{R}$ with $\xi_2 \geq \xi_1$ (see Figure 5.1), we have a monotonicity w.r.t. increasing ξ. This implies

$$\Omega(x) = \Omega(x, T) = [\underline{\xi}(x), \infty),$$

for some lower bound $\underline{\xi} : X \to \mathbb{R}$.

With this representation of the set of feasible realizations $\Omega(x)$, we can reformulate the probability evaluation as

$$\mathbb{P}(D(x,\delta)) = \mathbb{P}([\delta,\infty)) = 1 - F(\delta).$$

The result of the inner set-approximation problem with this design function is noted in Table 5.12.

Solution x^*	design variable δ^*	t_{total} [in sec]
$(1, 0.5728, -0.4272)$	-1.282	5.51

Table 5.12: Numerical results of set-approximation approach for SDC ellipse in uncertain triangle

Example: Circle in circle

We have already seen that the unique analytical solution of this problem for $R = 15, r = 3$ and $p = 0.9$ is

$$f(x^*) = -x^* \approx -5.592.$$

We start all discretization schemes with $x_0 := 4$ and $T_0 := \{(-1,0),(1,0),(0,-1),(0,1)\}$. Using the sandwiching approach based on the uniform discretization scheme, we get the following results listed in Table 5.13.

# points in		Approximations		time [in sec]		
$[-1,1]^2$	$B_1(0)$	$\underline{x}^*$	$\overline{x}^*$	$\underline{t}$	$\overline{t}$	t_{total}
$(1+2^1)^2 = 9$	5	5.768	5.438	3090	1.13	3090
$(1+2^2)^2 = 25$	13	5.776	5.446	15400	2.42	15400

Table 5.13: Iterates and computation times using a sandwiching approach with uniform discretization scheme

Please note that the calculation times are highly increased in this example as the random variable is two-dimensional and therefore the probability evaluations has to consider 50× as many samples as in the one-dimensional case.

Next to the uniform based approach, we are interested in the results of the LLAD and HLAD. Since the worst-case scenarios $(t^*(x,\xi))_{\xi\in\Xi}$ depend on the decision x and also on the realization of ξ, we cannot expect the LLAD to find the same points as HLAD. Furthermore, we can just expect to find approximations of the minimizer of the original problem because we have to approximate the set $S = \partial B_1(0)$. This cannot be done in finite many steps by discretization schemes. The iterates and computation times of the sandwiching approach based on the maximum structure using LLAD variant can be found in Table 5.14.

Iteration	Approximations		time [in sec]			
k	$\underline{x}^*$	$\overline{x}^*$	t_S	$\underline{t}$	$\overline{t}$	t_{total}
1	5.6343	5.4898	1.09	4060	4.69	4070
2	5.6313	5.4956	1.21	14900	6.88	14900

Table 5.14: Iterates and computation times using a sandwiching approach with the maximum structure using LLAD variant

Since we generate 10 random picks in each iteration per constraint in the maximum structure using LLAD variant, we created up to 10 worst-case scenarios in each iteration. As our problem is just defined by one inner function, the maximum structure using LLAD/HLAD variant and the LLAD/HLAD coincide in this example.
The results from the HLAD variant are presented in Table 5.15.

Iteration	new point	Approximations		time [in sec]			
k	t_k^*	$\underline{x}^*$	$\overline{x}^*$	t_S	$\underline{t}$	$\overline{t}$	t_{total}
1	$(-0.05321, 0.4905)$	5.781	5.450	505	4240	2.67	4740
2	$(0.03954, -0.6736)$	5.749	5.420	730	10600	23.4	11300

Table 5.15: Iterates and computation times using a sandwiching approach with the maximum structure using HLAD variant

After finding the discretization point $t_1^* = (-0.0532, 0.4905)$ in the first iteration step, the solver does not find a new discretization point at the boundary, but $t_2^* = (0.0395, -0.6736)$. This is again due to the poor analytical structure of $\varphi(x_1, T_1 \cup \{\cdot\})$ as in the last example.

Please note that in this example the outer approximations $\underline{x}^*$ are monotonically decreasing while the inner approximations $\overline{x}^*$ are monotonically increasing as we would expect using a sandwiching approach. Because a decision variable that ensures the feasibility of $x \geq 0$ is missing in this example, we do not find the optimal solution by solving the inner set-approximation problems directly.
For the set-approximation approach, we use the knowledge we gained by the analysis of the solution in the last section and define

$$\begin{aligned} &\Delta := [0, R], \\ &D : X \times \Delta \to \mathbb{R}, (x, \delta) \mapsto B_\delta(0). \end{aligned}$$

Because these designs are monotonically increasing (w.r.t. $\subseteq$) when we increase the radius δ this implies that given a decision $\overline{x} \in X$ there exists a radius $\delta(\overline{x}) \geq 0$ such that

$$\Omega(\overline{x}) = B_{\delta(\overline{x})}(0).$$

Therefore, we can reformulate the probability evaluation as

$$\mathbb{P}(D(x,\delta)) = \mathbb{P}(B_\delta(0)) = 1 - \exp\left(-\frac{\delta^2}{2}\right).$$

The solution of the corresponding inner set-approximation is given by Table 5.16.

Solution x^*	design variable δ^*	t_{total} in [sec]
5.592	2.146	0.918

Table 5.16: Numerical results of set-approximation approach for SDC circle in circle

Numerical experience

At the end of this section, we summarize our experience with stochastic design-centering problems and note the dos and don'ts handling them.
Before discussing the performance of the different solution approaches, we comment on the choice of numerical parameters to start the sandwiching algorithm 7.
Choice of a initial point x_0:

- We recommend to use a feasible initial point x_0 to start the algorithm. This can be calculated, e.g., by solving the probust optimization problem with objective function 0. We can then start the solution processes of the subproblems defined in line 4 and 6 of algorithm 7 by using the inner approximate $\overline{x}_k$ for all iterations $k \in \mathbb{N}$. This way, the used optimizer does not has to find the feasible set. This task alone might lead to its breakdown. Especially, if the initial point $x_0 \in X$ is either "too feasible" implying $\varphi(x, T_k) = 1$ for all x in a neighborhood of x_0 or "too infeasible" implying $\varphi(x, T_k) = 0$ for all x in such a neighborhood, the optimizer struggles.

- If we just use the probust subset algorithm instead of the sandwiching algorithm, we can use the last iterate to start the new optimization process. Nevertheless, it comes in handy to have a reference point x_0 that is feasible w.r.t. the original probust optimization problem such that we can reset the starting point in iterations $k \in \mathbb{N}$ where the optimizer started at x_k and could not find its way back into the feasible set $\mathcal{F}_{T_{k+1},p}$.

Choice of a initial discretization T_0:

- We recommend to use an initial discretization T_0 if we either know "interesting" scenarios for our solution beforehand or if the feasible set $\mathcal{F}_{T_0,p}$ is unbounded. In the first case, we save running time if we use adaptive discretization schemes. In the second case, we ensure that the optimizer finds a minimizer. Here, one has to balance the cardinality of T_0 to describe the set of feasible realizations well on the one hand and not to increase the numerical extra effort handling these additional scenarios in each step of the algorithm on the other hand. This numerical effort takes more than 95% of the running time and scales proportional to the considered size of T_0 as we can see, e.g., in Table 5.2.

Choice of a subset scheme (compare also the positioning of the schemes in Figure 4.1):

- We recommend to choose the *uniform discretization scheme* just to generate knowledge about the problem. With running times that pass the one hour mark after a few iterations, this scheme should be used wisely. Please note that these running times result from the calculation of an outer approximate. If we want to approximate some special scenario $t^* \in T$ as, e.g., in the first example, it might happen that we never exactly hit this point $t^* \in T_k$ for any $k \in \mathbb{N}$. On the contrary, if we cannot specify interesting scenarios and furthermore a whole interval of points seems interesting (see Example 1.1.9) this approach might be useful.
  Please note that in Table 5.1 we can see that the condition $\underline{x}_k = \underline{x}_{k+1}$ is not sufficient to stop the algorithm.

- We recommend to choose the *LLAD scheme* if we can calculate the induced maximizers in line 5 of algorithm 7 efficiently. As this is given for all examples in this chapter, the scheme performs well. It needs one to ten seconds - a fraction of the total running time - to generate new discretization points and generates outer approximates in a reasonable time interval compared to the uniform discretization scheme. Especially, when the worst-case scenarios are independent of the realization as in the first example. Then we can find the worst-case scenarios fast and at the same time guarantee the same solution precision as the HLAD (see Table 5.3). Drawbacks of this approach are that it takes random realizations in each iteration and therefore its results are not reproducible in general and that it might calculate worst-case scenarios for "unprobable" realizations. Furthermore, we do not know when to stop the algorithm if we do not find inner approximates $\overline{x}_k, k \in \mathbb{N}$ because the chosen $\xi \in \dot{\Xi}_k$ and therefore the corresponding $t(\underline{x}_k, \xi)$ have no structure in general.

- We recommend to choose the *HLAD scheme* if we can calculate the induced minimizers in line 5 of algorithm 7 efficiently. We have seen in the last two examples that finding this solution is numerically challenging even when the structure of the problems seems nice from a theoretical point of view. The HLAD then needs around ten seconds - a fraction of the total running time - to generate new discretization points and generates outer approximates quite fast compared to the uniform discretization and LLAD scheme. The HLAD benefits a lot from a warm-start by choosing x_0 and T_0 appropriately. It can happen that the HLAD gives back the starting scenario $t_0 \in T$ as its minimum altough it is already contained in T_k or obviously suboptimal. Recalling Lemma 2.3.4 we can stop the algorithm then because we have either found a solution $\underline{x}_k$ of the original probust optimization problem in the last iteration $k \in \mathbb{N}$ or we cannot find any new worst-case scenario $t_k^*(x) \in T$, e.g. because $\varphi(x_k, T_k \cup \{\cdot\})$ is locally constant around the initial scenario.

In the end, we want to comment on how to use analytical information to speed up the solver:

1. Replacing the probability evaluation function by an analytic function speeds of the process considerably as one can see comparing the set-approximation approaches with the sandwiching approaches. If such knowledge is given, it should be used.

2. A reduction of the search space influences also the evaluation time of the SRD. Fixing $x_2 = 0$ in the first example leads to calculating times that are just one third of the ones represented here.

3. If the inner function is a maximum of several inner functions, we recommend to use this structure as explained in Section 5.2. Otherwise, the optimizer calculating the worst-case scenarios in the "standard" LLAD or HLAD can have trouble finding the true worst-case of a maximum of differentiable functions and cancel prematurely.

Summary

In this chapter, we considered three problem instances of stochastic design-centering problems. We reformulated them as standard probust optimization problems and used the introduced theory from Chapter 1 and a combination of analytical and geometrical arguments to calculate their solution.
We then tested four of the solution approaches introduced in Part I of this theses on these problem instances and discussed their behavior.
In the following chapter, we consider a more realistic application. As we do not know the structure of the set of feasible realizations in this problem class, we use the numerical experience we gained in this chapter to define sandwiching algorithms that can solve the corresponding probust optimization problems efficiently.

6 Water reservoir problems

After testing the solution schemes for probust optimization problems that we introduced in Part I of this thesis in the last chapter, we want to use the gained insights to handle more realistic applications.
The main focus of this chapter is handling probust water reservoir problems, gaining insights about the problem structure and comparing the solutions induced by different uncertainty models.
We start this chapter by a short introduction how to model such water reservoir problems. Afterwards, we specify two problem instances in Section 6.1 which influence the design of a specified probust subset scheme in Section 6.2. We then solve the introduced problem instances and discuss their numerical results in Section 6.3.

A water reservoir problem (WR) in this thesis has the following form

$$\text{WR: } \max_{x \in X} f(x) \text{ s.t. } \underline{l} \leq l(x,t) \leq \bar{l} \; \forall t \in [0,T],$$

where $x \in X$ are operating variables to influence the water level of the reservoir given by $l : X \times [0,T] \to \mathbb{R}$ within the time horizon $[0,T], T \geq 0$, $\underline{l}, \bar{l} \in \mathbb{R}$ are lower and upper bounds for the water level and $f : X \to \mathbb{R}$ is some objective function.
One possible objective function is the total output $-\int_0^T x(t)dt$ over the time horizon $[0,T]$ of a measurable extraction profile $x : [0,T] \to \mathbb{R}_{\geq 0}$.
Because we want to guarantee that upper and lower water levels $\underline{l}, \bar{l}$ are respected at all time points $t \in [0,T]$ this leads to infinite many restrictions. If the extraction profile x is described by a finite dimensional vector, we study a semi-infinite optimization problem.
We often have an uncertain influence changing the water level by either an uncertain inflow, e.g., due to rain or melting water, or an uncertain outflow, e.g., due to the demand of water of other parties in a multipurpose water reservoir. Consequently, we want consider a stochastic component in our model of a water reservoir. We take account of that by formulating the probust water reservoir problem (PWR) as

$$\text{PWR: } \max_{x \in X} \mathbb{E}(f(x,\xi)) \text{ s.t. } \mathbb{P}(\underline{l} \leq l(x,\xi,t) \leq \bar{l} \; \forall t \in [0,T]) \geq p,$$

where ξ are the realizations of a random process Z that represents a random influence of the water level over time and $p \in [0,1]$ is a lower bound for the probability to respect the minimum and maximium water levels given the stochastic influence.
As WR can be interpreted as a semi-infinite optimization problem, PWR is a probust optimization problem and is the model of interest in this chapter.

Such water reservoir problems have been analysed by Prékopa et al. in [60] and by Henrion in [44]. More complex models can be found in papers from van Ackooij et al. [4] and Xu et al. [87]. We recommend the working paper by Dupacova et al. [28] and the monograph from Loucks et al. [55] for an overview of problem formulations and solving strategies.

In the referred literature a single reservoir or a systems of reservoirs is studied either to build a new reservoir or to manage an existing one. The reservoirs then can be used to handle floods (which corresponds to a time horizon of a year) or to generate hydro-power that should be sold (which corresponds to a time horizon of one day).

In either case, reservoir problems are described by at least one equation that links its water level over different time points $t_1, t_2 \in [0, T]$ by

$$l(x, \xi, t_1) = l(x, \xi, t_2) + I(x, \xi, t_1, t_2) - O(x, \xi, t_1, t_2), \tag{6.1}$$

where I, O are functions that describe the inflow and outflow into and out of the reservoir using the extraction profile $x \in X$ under realization ξ in the time interval $[t_1, t_2]$.

In the given literature this function is considered for a fixed, finite number of time points $t_1, ..., t_N \in [0, T]$ instead of the whole time interval $[0, T]$.

This reduces the induced problem to a chance constrained optimization problem. One way to solve this problem is to solve the constraints individually for any fixed time point. This leads to the constraint

$$\mathbb{P}(\underline{l} \le l(x, \xi, t) \le \bar{l}) \ge p \ \forall t = t_1, ..., t_N. \tag{6.2}$$

These constraints can often be handled deterministically by separating the realizations on one side of the inequality and using the corresponding cumulative density function. This approach evades numerically costly evaluations of the probability function.

If the constraints are handled as joint chance constraints, one often uses penalty methods for optimization and simulation techniques like Monte-Carlo simulation for probability evaluations to handle these problems.

In this work, we consider the whole time horizon $[0, T]$ instead of a discrete subset. On the contrary, we make some simplifying assumptions w.r.t. other modelling aspects such as the distribution of the random influence or the physical effects such as evaporation, spill or energy production which are neglected or strongly simplified.

We assume that the outflow O of the water reservoir is solely explained by our decision, while the inflow is solely described by a Gaussian process.

Therefore, we can write

$$l(x, \xi, t) = l_0 + \int_0^t \xi(\tau) - x(\tau) d\tau.$$

To handle the decision and the random process numerically, we approximate them by finite dimensional decision vectors $\tilde{x} \in \tilde{X} \subseteq \mathbb{R}^n$ and random vectors $\tilde{Z} : \Xi \to \mathbb{R}^m$.

After this short introduction to probust water reservoir problems, we define two specific problem instances in the next section.

6.1 Problem instances

As we introduced a model to describe water reservoir problems in a probust context, our goal in this section is to motivate two specific problem instances that are handled numerically in the last section of this chapter.

Linear objective with lower water level constraint

First, we consider a probust water reservoir problem where we are taking as much water out of a water reservoir as possible. To evade the trivial solution of letting the reservoir run full and then pump out all the water in the last time period, we also use a price signal $c \in L^2([0,T], \mathbb{R}_{\geq 0})$ to motivate water release at earlier time points. Given a lower water level $\underline{l} \geq 0$ and a probability threshold $p \in [0,1]$, we can note the probust water reservoir problem as

$$\begin{aligned} \mathrm{PWR}_1 : \max_{x \in L^2([0,T],\mathbb{R})} \int_0^T c(t)x(t)dt \text{ s.t. } & \mathbb{P}(\underline{l} \leq l(x,\xi,t) \ \forall t \in [0,T]) \geq p, \\ & x(t) \geq 0 \ \forall t \in [0,T]. \end{aligned}$$

Following the idea given in a paper by Berthold, Heitsch, Henrion and Schwientek in [15], we approximate the decision variable $x \in L^2([0,T], \mathbb{R}_{\geq 0})$ by a piece-wise constant function

$$x(t) \approx \sum_{i=1}^{n} \tilde{x}_i \chi_{[t_{i-1},t_i]}(t) \tag{6.3}$$

which can be described by the finite dimensional coefficient vector $\tilde{x} \in \mathbb{R}^n$ and a separation of $[0,T]$ into sub-intervals $T_i = [t_{i-1}, t_i]$ with $i = 1, ..., n$. These sub-intervals are defined by fixed time points $t_i \in [0,T]$ with $i = 0, ..., n$. With this approximation we can determine the outflow of the water reservoir by

$$O(t) = \int_0^t x(\tau)d\tau \approx \sum_{k=1}^{i^*(t)-1} x_k(t_k - t_{k-1}) + x_{i^*(t)}(t - t_{i^*(t)-1}), \tag{6.4}$$

where $i^*(t) = \min_{i \in \{1,...,n \ | \ t \leq t_i\}} i$ defines the index of $\tilde{x} \in \tilde{X}$ that represents the water extraction at the time $t \in T$.
Also we approximate the integral over the stochastic inflow by a superposition of sinus functions by

$$I(t) = \int_0^t \xi(\tau)d\tau \approx \sum_{j=1}^{m} \sin(\omega_j t)\tilde{\xi}_j + M(t),$$

where $M : [0,T] \to \mathbb{R}_{\geq 0}, t \to \int_0^t \mu(t)dt$ is defined by the integral of the expected value function $\mu : [0,T] \to \mathbb{R}_{\geq 0}$ of the Gaussian process modelling the random inflow.

Furthermore, $\omega_j > 0$ are inflow frequencies for $j = 1, .., m$ and $\tilde{\xi} \in \mathbb{R}^m$ is a realization of a normal distributed random variable $\tilde{Z} \sim \mathcal{N}\left(0, \tilde{\Sigma}\right)$ with covariance matrix $\tilde{\Sigma} \in \mathbb{R}^{m \times m}$. Recalling Equation (6.1), this leads for any $\tilde{x} \in \tilde{X}, \tilde{\xi} \in \mathbb{R}^m$ and some time point $t \in T$ to the constraint

$$\begin{aligned} g(\tilde{x}, \tilde{\xi}, t) &= \underline{l} - l(\tilde{x}, \tilde{\xi}, t) \\ &= \underline{l} - l_0 - \sum_{j=1}^{m} \tilde{\xi}_j \sin(\omega_j t) - M(t) + \sum_{k=1}^{i^*(t)-1} \tilde{x}_k (t_k - t_{k-1}) + \tilde{x}_{i^*(t)} (t - t_{i^*(t)-1}) \leq 0. \end{aligned}$$

With the accumulated price vector $\tilde{c} \in \mathbb{R}^n$ defined by

$$\tilde{c}_i = \int_{t_{i-1}}^{t_i} c(t) dt, i = 1, ..., n,$$

this implies the following reformulation of PWR_1 as a standard probust optimization problem

$$\begin{aligned} \text{PWR}_1\text{-SPP} : \min_{\tilde{x} \in \mathbb{R}^n} - \sum_{i=1}^{n} \tilde{c}_i \tilde{x}_i \text{ s.t. } & \mathbb{P}(g(\tilde{x}, \tilde{\xi}, t) \leq 0 \ \forall t \in [0, T]) \geq p, \\ & - \tilde{x}_i \leq 0 \ \forall i = 1, ..., n. \end{aligned}$$

Be aware that we can also represent the water level constraint based on the different extraction periods by the constraints

$$g_i(\tilde{x}, \tilde{\xi}, t) = \underline{l} - l_0 - \sum_{j=1}^{m} \tilde{\xi}_j \sin(\omega_j t) - M(t) + \sum_{k=1}^{i-1} \tilde{x}_k (t_k - t_{k-1}) + \tilde{x}_i (t - t_{i-1}) \leq 0,$$

with $\tilde{x} \in \tilde{X}, \tilde{\xi} \in \mathbb{R}^m, t \in T_i$ and $i = 1, ..., n$.
These constraints lead to the optimization problem

$$\begin{aligned} \widetilde{\text{PWR}_1}\text{-SPP} : \min_{\tilde{x} \in \mathbb{R}^n} - \sum_{i=1}^{n} \tilde{c}_i \tilde{x}_i \text{ s.t. } & \mathbb{P}(g_i(\tilde{x}, \tilde{\xi}, t) \leq 0 \ \forall t \in T_i, i = 1, ..., n) \geq p, \\ & - \tilde{x}_i \leq 0 \ \forall i = 1, ..., n. \end{aligned}$$

Given these reformulations of PWR_1, we can study its structure to decide which probust subset scheme might be useful to solve it.
We start with the set of scenarios. Not only $[0, T]$, but also its separation into smaller intervals T_i with $i = 1, ..., n$ allows us to consider compact, convex and one dimensional sets. Consequently, these sets are attractive to use as search spaces in optimization problems, e.g. by the maximum using variants of LLAD (see Equation (5.3)) or the HLAD (see Equation (5.4)).
In contrast to the structure of the search space, the objective function in these approaches is non-convex w.r.t. $t \in [0, T]$ since the functions g, g_i with $i = 1, ..., n$ are non-convex. Consequently, the usage of HLAD and LLAD is numerically challenging.

Nevertheless, given a fixed $i = 1, ..., n$ and $t \in T_i$, the inner function g_i is affine-linear w.r.t. $(\tilde{x}, \tilde{\xi})$ which implies a simple structure of the set of feasible realizations.
For a fixed $\tilde{x} \in \mathbb{R}^n$ we can express this set as

$$\Omega(\tilde{x}) = \bigcap_{i=1,..,n} \bigcap_{t \in T_i} \Omega_i(\tilde{x}, t),$$
$$\text{where } \Omega_i(\tilde{x}, t) = \{\tilde{\xi} \in \Xi \mid g_i(\tilde{x}, \tilde{\xi}, t) \leq 0\}.$$

Because the sets $\Omega_i(\tilde{x}, t)$ are by definition of g_i half-spaces for fixed $\tilde{x}$ and $t \in T$ for all $i = 1, ..., n$, we know that $\Omega(\tilde{x})$ is a convex set for all $\tilde{x} \in \mathbb{R}^n$. Furthermore, it is closed by the continuity of the function $g_i(\tilde{x}, \cdot, t)$ w.r.t. $\tilde{\xi} \in \mathbb{R}^m$ for fixed $i = 1, ..., n$, fixed $t \in T_i$ and $\tilde{x} \in \tilde{X}$.
To ensure the boundedness of the set of feasible realizations, we consider the following statement:

Proposition 6.1.1 (The sets of feasible realizations of PWR$_1$-SPP are bounded) *Given an extraction profile $\tilde{x} \in \tilde{X}$, $T > 0$, an initial water level $l_0 \geq \underline{l}$, a non-negative expected value function $\mu \in L^2([0, T], \mathbb{R}_{\geq 0})$ of the considered random process and frequencies $\omega_1, ..., \omega_m \geq 0$ with $m \in \mathbb{N}$ that satisfy $\omega_1 \geq \frac{2\pi}{T}$ and $\omega_{j+1} \geq 2\omega_j$ for all $j = 1, ..., m-1$, then the set of feasible realizations $\Omega(\tilde{x})$ of PWR$_1$-SPP is bounded.*

Proof. We show the claim indirectly:
We assume that there exists a sequence $(\xi_k)_{k \in \mathbb{N}} \subseteq \Xi$ such that $\lim_{k\to\infty} ||\xi_k||_\infty = \infty$ and $\max_{t \in T} g(\tilde{x}, \xi_k, t) \leq 0$ for all $k \in \mathbb{N}$ and a given $\tilde{x} \in \tilde{X}$. Consequently, 0 is an upper bound of the sequence $(g(\tilde{x}, \xi_k, t))_{k \in \mathbb{N}}$ for any $t \in [0, T]$.
To show that this sequence is unbounded, we represent g for fixed $\tilde{x} \in \tilde{X}$ and for fixed $t \in [0, T]$ as an affine-linear function w.r.t. $\tilde{\xi} \in \mathbb{R}^m$ and then focus on special scenarios.
We know by the definition of g that

$$\begin{aligned} g(\tilde{x}, \tilde{\xi}, t) &= \underline{l} - l_0 - \sum_{j=1}^{m} \tilde{\xi}_j \sin(\omega_j t) - M(t) + \sum_{k=1}^{i^*(t)-1} \tilde{x}_k (t_k - t_{k-1}) + \tilde{x}_{i^*(t)} (t - t_{i^*(t)-1}) \\ &= \sum_{j=1}^{m} w_j(t) \tilde{\xi}_k + v(\tilde{x}, t) \end{aligned}$$

holds for $\tilde{x} \in \tilde{X}$ and any $t \in [0, T]$, where we used

$$\begin{aligned} &w_j(t) = -\sin(\omega_j t), j = 1, ..., m \text{ and} \\ &v(\tilde{x}, t) = \underline{l} - l_0 - M(t) + \sum_{k=1}^{i^*(t)-1} \tilde{x}_k (t_k - t_{k-1}) + \tilde{x}_{i^*(t)} (t - t_{i^*(t)-1}). \end{aligned}$$

Given $\tilde{x} \in \tilde{X} \subseteq \mathbb{R}^n_{\geq 0}$, as well as $l_0 \geq \underline{l}$ and $M(t) = \int_0^t \mu(t)dt \geq 0$ by assumption, we can estimate $v(\tilde{x}, t)$ for any fixed $t \in [0, T]$ by:

$$\begin{aligned}
v(\tilde{x}, t) &= \underline{l} - l_0 - M(t) + \sum_{k=1}^{i^*(t)-1} \tilde{x}_k(t_k - t_{k-1}) + \tilde{x}_{i^*(t)}(t - t_{i^*(t)-1}) \\
&\geq \underline{l} - l_0 - M(T) + \sum_{k=1}^{n} \tilde{x}_k(t_k - t_{k-1}) \\
&\geq \underline{l} - l_0 - M(T) =: C
\end{aligned}$$

With this estimation for $v(\tilde{x}, t)$, we can estimate for any $k \in \mathbb{N}$ and $t \in [0, T]$:

$$\begin{aligned}
&- \sum_{j=1}^{m} \sin(\omega_j t)\xi_{j,k} + C \leq g(\tilde{x}, \xi_k, t) \leq 0 \\
\Rightarrow &- \sum_{j=1}^{m} \sin(\omega_j t)\xi_{j,k} \leq -C
\end{aligned}$$

By our assumptions we know that $\omega_1 \geq \frac{2\pi}{T}$ and $\omega_{j+1} \geq 2\omega_j$ for all $j = 1, ..., m-1$. Consequently, there exists a $t_k^* \in [0, T]$ such that $\sum_{j=1}^m - \sin(\omega_j t_k^*)\xi_{j,k} \geq \max_{j=1,...,m} |\xi_{j,k}|$ for any fixed $k \in \mathbb{N}$. Therefore, the last inequality states for this fixed t_k^*

$$||\xi_k||_\infty = \max_{j=1,...,m} |\xi_{j,k}| \leq - \sum_{j=1}^{m} \sin(\omega_j t_k^*)\xi_{j,k} \leq -C$$

Because we assumed that $\lim_{k\to\infty} ||\xi_k||_\infty = \infty$, we can ensure that this last inequality does not hold for $k \to \infty$. Therefore, the claim holds. □

With this statement we can guarantee that the set of feasible realizations $\Omega(\tilde{x})$ is a convex, compact set for all $\tilde{x} \in \tilde{X}$.

In the next step, we consider the feasible set of PWR$_1$-SPP. By the definition g and due to Proposition 1.1.6, we can ensure that the probust function $\varphi : \tilde{X} \to [0, 1]$ is upper semi-continuous and therefore, the feasible set is closed. Moreover, Proposition 1.1.8 implies that the feasible set is convex. To show that this set is bounded, we need another statement:

Proposition 6.1.2 (The feasible set of PWR$_1$-SPP is bounded)
Given $T > 0$ and $p \in (0, 1]$, then the feasible set of PWR$_1$-SPP is bounded.

Proof. We show the claim indirectly:
We assume that there exists a sequence $(x_k)_{k\in\mathbb{N}} \subseteq \tilde{X}$ such that $\lim_{k\to\infty} ||x_k||_1 = \infty$ and $\varphi(x_k) \geq p$ for all $k \in \mathbb{N}$. Consequently, p is a lower bound of the sequence $(\varphi(x_k))_{k\in\mathbb{N}}$. We show that for each $r > 0$, we can find a $N \in \mathbb{N}$ such that the feasible sets $\Omega(x_k)$ does not contain any realization from $B_r(0)$ for all $k \geq N$. As the set $B_r(0)$ is monotonically increasing w.r.t. $\subseteq$ for increasing $r > 0$, this contradicts our basic assumption $\varphi(x_k) \geq p$ for all $k \in \mathbb{N}$, when fixing some $R > 0$ such that $B_R(0)$ satisfies $\mathbb{P}(B_R(0)) > 1 - p$.

Let us fix that $R > 0$ as well as any $\overline{\xi} \in B_R(0)$. If $\overline{\xi}$ is a feasible realization for $\tilde{x} \in \tilde{X}$, then it satisfies the probust constraint especially for $t = T$, what implies

$$g(\tilde{x}, \overline{\xi}, T) = \underline{l} - l_0 - \sum_{j=1}^{m} \sin(\omega_j T)\overline{\xi}_j - M(T) + \sum_{i=1}^{n} \tilde{x}_i (t_i - t_{i-1}) \leq 0. \tag{6.5}$$

Because we can estimate

$$\begin{aligned} \underline{l} - l_0 - \sum_{j=1}^{m} \sin(\omega_j T)\overline{\xi}_j - M(T) &\geq \underline{l} - l_0 - \sum_{j=1}^{m} |\overline{\xi}_j| - M(T) \\ &\geq \underline{l} - l_0 - \sum_{j=1}^{m} \overline{r} - M(T) =: C, \end{aligned}$$

this implies with inequality (6.5):

$$\begin{aligned} &\sum_{i=1}^{n} \tilde{x}_i (t_i - t_{i-1}) + C \leq g(\tilde{x}, \overline{\xi}, T) \leq 0 \\ \Rightarrow &\min_{i=1,\dots,n} (t_i - t_{i-1}) \sum_{i=1}^{n} \tilde{x}_i = \min_{i=1,\dots,n} (t_i - t_{i-1}) ||\tilde{x}||_1 \leq -C \end{aligned}$$

Since $\lim_{k\to\infty} ||x_k||_1 = \infty$, there exists a $N \in \mathbb{N}$ such that the last inequality is violated for all $k \geq N$.
Because we chose $\overline{\xi} \in B_R(0)$ arbitrarily and C does not depend on $\overline{\xi}$, we can ensure

$$B_R(0) \cap \Omega(x_k) = \emptyset$$

for all $k \geq N$. This implies that all $x_k, k \geq N$ are infeasible and therefore the feasible set of PWR$_1$-SPP is bounded. □

Since our objective function is linear w.r.t. $\tilde{x} \in \tilde{X}$, we know that PWR$_1$-SPP is a convex optimization problem.
In Section 6.3, we consider a special problem instance induced by the following parameters:

$$n = 24, m = 10, T = 24, \tag{6.6}$$

$$\omega_1 = \frac{31\pi}{420}, \omega_j = \sqrt{5}\omega_{j-1} \ \forall j = 2, .., 10, \tag{6.7}$$

$$\underline{l} = 0.25, l_0 = 0.3, \tag{6.8}$$

$$p = 0.9, \mu \equiv 0.1, \tilde{Z} \sim \mathcal{N}\left(\tilde{\mu}, \tilde{\Sigma}\right), \tag{6.9}$$

$$\tilde{\mu} = 0_{\mathbb{R}^{10}}, \tilde{\Sigma} = diag(\sigma_1^2, ..., \sigma_{10}^2), \tag{6.10}$$

$$\sigma_1 = 0.06, \sigma_j = \frac{\sigma_{j-1}}{2} \ \forall j = 2, ..., 10. \tag{6.11}$$

Approximated energy production objective, two water level constraints with additional constraints

Next to PWR$_1$ which aims for extracting as much water as possible given an abstract cost function and without any additional constraints, the second problem instance focuses on more details. The corresponding objective function measures the amount of energy produced by the extracted water. This function is approximated by the amount of water we take out of the reservoir multiplied by the water level like in the paper of Andrieu et al. [7]. Additionally, we are interested in an upper and lower water level which should be respected in a given percentage of inflow cases as well as some minimal and maximal outflow constraints and a so called cycling constraint. Thereby, the cycling constraint guarantees that the water level at $t = T$ is expected to be at least as high as at $t = 0$. This leads to the following problem:

$$\begin{aligned} \mathrm{PWR}_2 : \max_{x \in L^2([0,T],\mathbb{R})} \mathbb{E}\left(\int_0^T x(t) l(x,\xi,t) dt \right) \text{ s.t. } & \mathbb{P}(\underline{l} \leq l(x,\xi,t) \leq \bar{l} \ \forall t \in [0,T]) \geq p, \\ & x_i(t) \in [\underline{x}(t), \overline{x}(t)] \ \forall t \in [0,T] \qquad (6.12) \\ & \mathbb{E}(l(x,\xi,T)) \geq l_0. \qquad (6.13) \end{aligned}$$

To simplify this problem, we use the same approximations of the extraction profile x and the random inflow realizations ξ as in the last problem instance, see (6.3) and (6.4). Denoting these approximations again by $\tilde{x} \in \mathbb{R}^n_{\geq 0}$ and $\tilde{\xi} \in \mathbb{R}^m$, we reformulate PWR$_2$ as a probust optimization problem.
We start with reformulating the cycling constraint (6.13):

$$\begin{aligned} \mathbb{E}\left(l(\tilde{x}, \tilde{\xi}, t) \right) &= \mathbb{E}\left(l_0 + \sum_{j=1}^m \tilde{\xi}_j \sin(\omega_j t) + M(t) - \sum_{i=1}^{i^*(t)-1} \tilde{x}_i (t_i - t_{i-1}) - \tilde{x}_{i^*(t)} (t - t_{i^*(t)-1}) \right) \\ &= l_0 + \sum_{j=1}^m \mathbb{E}(\tilde{\xi}_j) \sin(\omega_j t) + M(t) - \sum_{i=1}^{i^*(t)-1} \tilde{x}_i (t_i - t_{i-1}) - \tilde{x}_{i^*(t)} (t - t_{i^*(t)-1})) \\ &= l_0 + M(t) - \sum_{i=1}^{i^*(t)-1} \tilde{x}_i (t_i - t_{i-1}) - \tilde{x}_{i^*(t)} (t - t_{i^*(t)-1})) \end{aligned}$$

This means that the expected water level at $t = T$ is at least l_0 if and only if we pump out water that is equal or less then the expected amount of water inflow in this time interval:

$$\begin{aligned} &\mathbb{E}\left(l(\tilde{x}, \tilde{\xi}, T) \right) \geq l_0 \\ \Leftrightarrow & M(T) \geq \sum_{i=1}^n \tilde{x}_i (t_i - t_{i-1}) \end{aligned}$$

Using this expression, we can also reformulate the objective function as:

$$\begin{aligned}
f(\tilde{x}) &= -\mathbb{E}\left(\int_0^T \sum_{i=1}^n \tilde{x}_i \chi_{[t_{i-1},t_i)}(t) l(\tilde{x},\tilde{\xi},t)dt\right) \\
&= -\int_0^T \sum_{i=1}^n \tilde{x}_i \chi_{[t_{i-1},t_i)}(t)\mathbb{E}(l(\tilde{x},\tilde{\xi},t))dt \\
&= -\sum_{i=1}^n \tilde{x}_i \int_{t_{i-1}}^{t_i} \mathbb{E}(l(\tilde{x},\tilde{\xi},t))dt \\
&= -\sum_{i=1}^n \tilde{x}_i \int_{t_{i-1}}^{t_i} \left(l_0 + M(t) - \sum_{k=1}^{i-1} \tilde{x}_k(t_k - t_{k-1}) - \tilde{x}_i(t-t_{i-1})\right) dt \\
&= -\sum_{i=1}^n \tilde{x}_i \left(l_0(t_i - t_{i-1}) + \int_{t_{i-1}}^{t_i} M(t)dt - \sum_{k=1}^{i-1} \tilde{x}_k(t_k - t_{k-1})(t_i - t_{i-1})\right. \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\left. - \tilde{x}_i \frac{(t_i - t_{i-1})^2}{2}\right) \\
&= -\frac{T}{n}\sum_{i=1}^n \tilde{x}_i \left(l_0 + \frac{T}{n}\left(\frac{(2i-1)\mu}{2} - \sum_{k=1}^{i-1}\tilde{x}_k - \frac{\tilde{x}_i}{2}\right)\right) \\
&= -\frac{T}{n}\left(\langle \tilde{x}, A\tilde{x}\rangle + \sum_{i=1}^n b_i \tilde{x}_i\right) \qquad (6.14)
\end{aligned}$$

This representation allows us to evaluate the objective function and its gradients w.r.t. $x \in X$ more easily, where we used

$$A = -\frac{T}{n}\begin{pmatrix} 0.5 & 0 & ... & 0 \\ 1 & 0.5 & ... & 0 \\ ... & & & \\ 1 & 1 & ... & 0.5 \end{pmatrix} \quad \text{and}$$

$$b = \left(l_0 + \frac{T}{n}\frac{(2i-1)\mu}{2}\right)_{j=1,...,n}.$$

Furthermore, we decompose the water level constraint into two inner functions of the form:

$$\begin{aligned}
g_1(\tilde{x},\tilde{\xi},t) &:= \underline{l} - l(\tilde{x},\tilde{\xi},t) \leq 0, \\
g_2(\tilde{x},\tilde{\xi},t) &:= g_1(\tilde{x},\tilde{\xi},t) - \overline{l} + \underline{l} \leq 0
\end{aligned}$$

Consequently, we can state the following standard probust optimization problem

$$\begin{aligned}
\mathrm{PWR}_2 - \mathrm{SSP}\colon \min_{\tilde{x}\in\mathbb{R}^n} f(\tilde{x}) \text{ s.t. } &\mathbb{P}\begin{pmatrix} g_1(\tilde{x},\tilde{\xi},t) \leq 0 \\ g_2(\tilde{x},\tilde{\xi},t) \leq 0 \end{pmatrix} \forall t \in [0,T] \geq p, \\
&\tilde{x}_i \in [\underline{x}_i, \overline{x}_i] \ \forall i = 1,...,n, \\
&M(T) \geq \sum_{i=1}^n \tilde{x}_i(t_i - t_{i-1}).
\end{aligned}$$

In Section 6.3, we consider the problem instance defined by fixed

$$n = 24, m = 10, T = 24, \tag{6.15}$$

$$\underline{x}_i = \begin{cases} 0.1, & \text{if } i \in \{6,7,8,20,21,22,23\} \\ 0, & \text{otherwise} \end{cases}, \tag{6.16}$$

$$\overline{x}_i = 0.2 \ \forall i = 1, ..., 24, \tag{6.17}$$

$$\omega_1 = \frac{7\pi}{80}, \omega_j = \sqrt{5}\omega_{j-1} \ \forall j = 2, .., 10, \tag{6.18}$$

$$\underline{l} = 0.25, l_0 = 0.3, \overline{l} = 1, \tag{6.19}$$

$$p = 0.9, \mu \equiv 0.1, \tilde{Z} \sim \mathcal{N}\left(\tilde{\mu}, \tilde{\Sigma}\right), \tag{6.20}$$

$$\tilde{\mu} = 0_{\mathbb{R}^{10}}, \tilde{\Sigma} = diag(\sigma_1^2, ..., \sigma_{10}^2), \tag{6.21}$$

$$\sigma_1 = 0.06, \sigma_j = \frac{\sigma_{j-1}}{2} \ \forall j = 2, ..., 10. \tag{6.22}$$

Because the objective function of PWR$_2$ decreases for increasing water levels l, a simple strategy would be to let the reservoir run full and then start pumping out such that the reservoir is not surpassing the upper water level (see Figure 6.11). This would imply a lot of worst-case time points created by the upper water level constraint. To disturb this strategy, we assume that we need some of the water in the early and late hours of the day (see Equation (6.16)). This is motivated by the idea that people need water at home such that a certain output is required.

6.2 Modified subset schemes

After describing the probust optimization problem instances in the last section, we have to choose an appropriate subset scheme to use with the sandwiching-algorithm 7. Recalling Figure 4.1, we note that we are unable to predefine worst-case time points according to a given extraction profile and an inflow realization analytically. In contrast to this gap of knowledge, we are aware that the set of feasible realizations is a convex set for all $\tilde{x} \in \mathbb{R}^n$. While the first aspect motivates to use an uniform discretization scheme, the second perspective supports the HLAD scheme. Therefore, we try to combine these two approaches in this section in a way that is numerically reasonable.

We do not use the LLAD, because the non-convexity of the inner functions of the probust water reservoir problem instances w.r.t. $t \in [0, T]$ makes it as difficult to evaluate as the new discretization scheme, while the new discretization scheme calculates determined time points in contrary to the randomly picked time points of the LLAD scheme which makes the interpretation of the results easier.

The new discretization approach closes the gap between universality of applicability of a discretization scheme and numerical stability and also is able to add user-dependent inputs to define the "importance of a scenario". We recommend using the following variant of the HLAD (see Equation (2.6)) that is inspired by a semi-infinite discretization by Reemtsen [64]

$$\Phi_k : X \times 2^T \to 2^T, (x, S) \mapsto S \cup \{t_k^*\}, \tag{6.23}$$

where $t_k^* \in \arg\min_{t \in G_{d_{j(k)}}} \varphi(x, S \cup \{t\})$ is the worst-case scenario in the grid $G_{d_{j(k)}}$ with grid size $d_{j(k)} > 0$ considering the probability evaluation for the current decision and added to the current scenario set S.

The set $G_d := \{t \in T \mid \exists j \in \mathbb{Z}^q : t = d \sum_{i=1}^q j_i e_i\}$ is the intersection of T with an uniform grid of size $d > 0$. We assume in the remainder of this section that the sequence $(d_j)_{j \in \mathbb{N}}$ fulfills $d_j > 0$ for all $j \in \mathbb{N}$ and $\lim_{j \to \infty} d_j = 0$.

To connect the grid size d_j with the iteration k of the probust subset scheme, we define another sequence $(\epsilon_j)_{j \in \mathbb{N}}$ with $\epsilon_j > 0$ for all $j \in \mathbb{N}$ and $\lim_{j \to \infty} \epsilon_j = 0$.

We start in iteration $k = 1$ with $j = 1$. Afterwards, we increase the index j by 1 if either we already took the whole grid into consideration or no remaining grid point is "important enough". These conditions can be formalized as

$$\begin{aligned} &G_{d_j} \subseteq T_k \text{ or} \\ &\varphi(x_k, T_k) - \varphi(x_k, T_k \cup \{t_k^*\}) \le \epsilon_j. \end{aligned}$$

We call this scheme the uniform-HLAD (UHLAD) scheme. By definition it is an adaptive increasing subset scheme. If $|T_0| < \infty$, it is a discretization scheme.

Before we concentrate on the convergence of UHLAD, we need the following useful proposition:

Proposition 6.2.1 ($j(k) \to \infty$ for $k \to \infty$)
Assume that $T \subseteq \mathbb{R}^q$ is compact and that $\varphi(x, S)$ is well-defined for all $x \in X, S \subseteq T$. Then it holds for the index $j(k), k \in \mathbb{N}$ defined in the UHLAD:

$$\lim_{k\to\infty} j(k) = \infty$$

Proof. We show the claim arguing indirectly:
Assume that $(j(k))_{k\in\mathbb{N}}$ is bounded from above. Because $j(k)$ is monotonically increasing by definition, there has to be an upper bound $N \in \mathbb{N}$ satisfying $j(k) \leq N$ for all $k \in \mathbb{N}$. Because T is compact, the set G_d is finite for all $d > 0$. Consequently, after at most $K = \sum_{j=1}^{N} |G_{d_j}| + 1$ iterations, we have collected more discretization points than the first N grids can offer and therefore have to increase j at least $N + 1$ times implying $j(K) \geq N + 1$. This contradicts that N is an upper bound for all $k \in \mathbb{N}$. Hence, the assumption that $(j(k))_{k\in\mathbb{N}}$ is bounded from above is wrong and the claim holds. □

With this proposition, we can show the convergence of the probust discretization scheme that uses UHLAD.

Lemma 6.2.2 (Convergence of UHLAD)
Choosing $\Phi = (\Phi_k)_{k\in\mathbb{N}}$ as the UHLAD with $(d_k)_{k\in\mathbb{N}}$ converging to zero, where the inner function $g : X \times \Xi \times T \to \mathbb{R}$ is a continuous and fulfilling Assumption 2.1.1 *for all $x \in X$ and T has no isolated points, then any accumulation point $\overline{x}$ of $(x_k)_{k\in\mathbb{N}}$ is an optimal solution of the original probust optimization problem.*

Proof. We prove this claim by using Theorem 2.2.3:
We show that the candidate-condition is fulfilled for any accumulation point $\overline{x} \in X$ of $(x_k)_{k\in\mathbb{N}}$ indirectly.
Please note that with by the compactness of T the set G_d is finite for all $d > 0$ and consequently the discretization scheme is well-defined for all $k \in \mathbb{N}$. Because T has no isolated points and $\lim_{k\to\infty} d_j = 0$, there exists some iteration $N \in \mathbb{N}$ such that $T \cap G_{d_k} \neq \emptyset$ for all $k \geq N$. Furthermore, due to the compactness of $X \subseteq \mathbb{R}^n$, the sequence $(x_k)_{k\in\mathbb{N}}$ has at least one accumulation point. Without loss of generality we assume that the whole sequence $(x_k)_{k\in\mathbb{N}}$ has just one accumulation point. Otherwise we switch notations to consider any (fixed) converging subsequence.

We fix the accumulation point $\overline{x} \in X$, set $S := \lim_{k\to\infty} T_k$ and assume

$$\exists t_f \in T, \epsilon > 0 : \varphi(\overline{x}, S) - \varphi(\overline{x}, S \cup \{t_f\}) \geq \epsilon.$$

This implies that there exists a measurable set $\Omega \in \mathcal{A}$ such that

$$\mathbb{P}(\Omega) \geq \epsilon \text{ and}$$
$$g(\overline{x}, \xi, t_f) > 0 \ \forall \xi \in \Omega.$$

Because $\mathbb{P}(\Omega) \geq \epsilon > 0$, there exists at least one element $\overline{\omega} \in \Omega$. Since $g(\overline{x}, \overline{\omega}, \cdot)$ is continuous w.r.t. $t \in T$, we know that there exists a radius $r > 0$ such that

$$g(\overline{x}, \overline{\omega}, t) > 0 \; \forall t \in B_r(t_f) \cap T.$$

By Proposition 6.2.1 and $\lim_{k \to \infty} d_{j(k)} = 0$, we know that there exists a $N \in \mathbb{N}$ such that for all $k \geq N$ we can guarantee that there exists a $t_k \in B_r(t_f) \cap G_{d_{j(k)}}$.
As $t_k \in B_r(t_f)$ is no element of S, we know that t_k is not added to T_k for any $k \in \mathbb{N}$. By definition of Φ_k this implies for any $k \in \mathbb{N}$

$$\varphi(x_k, T_k) - \varphi(x_k, T_k \cup \{t_k\}) < \epsilon_{j(k)}.$$

Taking the limes on both sides of the inequality and using $\lim_{k \to \infty} \epsilon_{j(k)} = 0$ as well as $\lim_{k \to \infty} t_k = t_f$ leads to

$$\lim_{k \to \infty} \varphi(x_k, T_k) - \varphi(x_k, T_k \cup \{t_k\}) = \varphi(\overline{x}, S) - \varphi(\overline{x}, S \cup \{t_f\}) \leq \lim_{k \to \infty} \epsilon_{j(k)} = 0.$$

Please note that we used in the last estimation

$$\begin{aligned}\lim_{k \to \infty} |\varphi(x_k, T_k \cup \{t_k\}) - \varphi(\overline{x}, S \cup \{t_f\})| \leq & \lim_{k \to \infty} \underbrace{|\varphi(x_k, T_k \cup \{t_k\}) - \varphi(\overline{x}, S \cup \{t_k\})|}_{\to 0 \text{ by Proposition 2.1.18}} \\ & + \lim_{k \to \infty} \underbrace{|\varphi(\overline{x}, S \cup \{t_k\}) - \varphi(\overline{x}, S \cup \{t_f\})|}_{\to 0 \text{ by Proposition 2.1.8}} \\ =& 0.\end{aligned}$$

Consequently, the candidate-condition is fulfilled and the claim holds. □

With this subset scheme for a fixed $x \in X$, we can not only "scan" the uncertainty set T for points that are "important enough" to reduce the probability evaluation at least by ϵ, but we can also decide how many points we want to evaluate and therefore control the calculation time by the grid size d.
Although the UHLAD seems to be less elegant than the HLAD from the theoretical point of view, it does handle the numerical downsides of the HLAD which we already discussed in Section 5.3 by additional computational effort. This makes the UHLAD useful to highlight the most important scenarios given a fixed decision $x \in X$ in an numerically stable, deterministic way. Therefore, it provides insights in the structure of worst-case scenarios over all possible realizations and is quite useful when handling probust optimization problems whose worst-case scenarios are not known analytically.

To make the new discretization scheme run in a reasonable time, we need to speed up the probability evaluation based on the SRD used in the last chapter. We modify our implementation as described in the paper of Berthold, Heitsch, Henrion and Schwientek [15] that showed significant improvements of calculation times:

1. We fix the sample of unit sphere vectors $\mathbb{S}^{m-1}$ that is used to evaluate the probability by the SRD in a problem instance. Therefore, we have a deterministic approximation of probability evaluation instead of a random approximation and are able to reuse calculated values w.r.t. the sample $\{v_1, ..., v_N\} \subseteq \mathbb{S}^{m-1}$.

2. We save the minimal ray lengths r_k that correspond to the given unit vectors v_k for each scenario $t \in T_k$ and all $k = 1, ..., N$ when calculating new time points by UHLAD. If we add a new discretization point t^* to T_k, we just have to calculate the corresponding ray lengths for $v_k, k = 1, ..., N$ and check if it is smaller than r_k. It is not useful to save the ray length for switching decisions because the decision affect the length of all rays $r_k, k = 1, ..., N$ in general.

3. We use the affine-linear structure of the water level constraint w.r.t. the inflow realizations $\xi \in \mathbb{R}^m$ to calculate the interception point between rays and the set of feasible realizations (see Equation (1.3)) faster.

4. We start the probust subset schemes with a softened stopping criterion under consideration of Theorem 1 in [15] that allows us to calculate suboptimal iterates. As we do not expect to converge towards the solution in the first few iterations and therefore do not need to calculate the corresponding iterates precisely. Doing so, we reduce the necessary probability evaluations until we stop. Over the iterations, we have to resharpen the solution precision again to guarantee optimality in the limes. A useful heuristic value how to sharpen the stopping criterion is $\varphi(x_k, T_k) - \varphi(x_k, T_{k+1})$.

6.3 Numerical results and comparison of uncertainty models

After defining the probust water reservoir problem instances and the solution methods in the last sections, we want to solve these problem instances. We start with two smaller instances related to PWR_1 and discuss their solutions to get a feeling how the objective function influences the choice of an optimal decision x^* which influences the set of feasible realizations and worst-case time points.
Afterwards, we solve the problem instances PWR_1-SPP and PWR_2-SPP introduced in Section 6.1 and compare the results of the following stochastic models:

1. The probust model which guarantees to stay always in between the critical water levels for a high percentage of inflow realizations. This model leads to a probust optimization problem solved by UHLAD.

2. The robust-probabilistic (later called robubilistic) model which guarantees to stay in between the critical water levels for any fixed time-point for a high percentage of inflow realizations. This model leads to a semi-infinite optimization problem that is solved by the adaptive discretization approach from Blankenship and Falk (see [17]).

3. The expected value model which guarantees that the expected inflow stays in between the critical water levels for all time-points. This model leads to a semi-infinite optimization problem that is also solved by the adaptive discretization approach from Blankenship and Falk (see [17]).

Probust solution for PWR_1-SPP

We consider two problem instances of PWR_1 using the data (6.7)-(6.11), but approximating the outflow and inflow by $n = m = 2$ dimensional variables and different objective functions. Thereafter, we solve the problem instance with $n = 24, m = 10$. By solving the smaller problems, we get a feeling for the new solution approach UHLAD, the worst-case time points, the set of feasible realizations and minimizing extraction profiles.
The first smaller problem instance is defined by the price signal $\tilde{c} = (1, 0)$. This means that we are interested in pumping out as much water as possible in the first time interval $[0, 12]$, while we just adapt the outflow in the second time interval $[12, 24]$ in a way that we satisfy the given constraints.
Consequently, the objective in this case is $f_1(\tilde{x}) = -\tilde{x}_1$. We start the solving process using UHLAD in Algorithm 7 with $\tilde{x}_0 = (0.05, 0.05), T_0 = \{0, 12, 24\}$, a grid of size $d_1 := 1$ and a precision parameter of $\epsilon_1 := 10^{-6}$. We choose 10 additional time points in each iteration to increase the set of worst-case time points and afterwards let the optimizer take 5 steps to calculate a new extraction profile. The corresponding iterates $\underline{x}_k, \overline{x}_k$ and calculation times are listed in Table 6.1 and Table 6.2.

iteration k	# point in $T_k \subseteq [0,24]$	$\underline{x}_k^*$	$\overline{x}_k^*$
1	13	$(0.09262, 0.05)$	infeasible
2	23	$(0.09058, 0.05)$	$(0.01493, 0)$
3	33	$(0.09058, 0.05)$	$(0.01493, 0)$
4	43	$(0.09058, 0.05)$	$(0.01493, 0)$
5	53	$(0.09058, 0.05)$	$(0.01493, 0)$
10	103	$(0.09057, 0.05)$	$(0.09056, 0)$
15	153	$(0.09057, 0.05)$	$(0.09056, 0)$

Table 6.1: Iterates of UHLAD sandwiching approach for PWR_1 instance with $\tilde{c} = (1, 0)$

iteration	# point in	time [in sec]			
k	$T_k \subseteq [0,24]$	t_{S}	$\underline{t}$	$\overline{t}$	t_{total}
1	13	54.5	36.1	infeasible	90.6
2	23	37.6	25.2	1.47	64.3
3	33	89.8	27.7	2.42	120
4	43	163	36.4	4.00	203
5	53	185	43.0	5.76	234
10	103	412	85.1	22.1	519
15	153	777	132	46.4	955

Table 6.2: Computation times of UHLAD sandwiching approach for PWR_1 instance with $\tilde{c} = (1, 0)$

We note that the first component of the decision is fixed fastly, because the main necessary discretization points are found in early iterations $k = 1, 2$. The optimal outflow in $[0, 12]$ seems to be roughly 0.906 what is around 10% less than the expected inflow in this time interval. These 10% can be interpreted as the buffer to not violate the lower water level constraint for too many realizations.
The set-approximation problem to determine $\overline{x}_k$ needs considerably more discretization points because $\Omega(x_k)$ has to cover the set $\Omega(x_k, T_k)$ which is unbounded for the first few iterations as depicted in Figure 6.2 a). As $\Omega(\tilde{x})$ is bounded by Proposition 6.1.1 and Proposition 3.1.7, we cannot find a feasible solution.
Because $f(\underline{x}_k)$ and $f(\overline{x}_k)$ do converge to each other, the algorithm can be stopped. Please note, that the minimizer of this problem is not unique since we are not interested in how much water is pumped out of the reservoir in the second time interval $[12, 24]$ as long as all constraints are satisfied.

Considering the calculation times, we see in Table 6.2 that the time to calculate new discretization points t_S increases over the iterations significantly. That is because the search space G_d grows exponentially with increasing $j(k) \in \mathbb{N}$ over the iterations $k \in \mathbb{N}$. The calculation time $\underline{t}$ of the iterate $\underline{x}_k$ is lower than t_S in all iterations. This signifies that the initial grid size d_1 was chosen to be to fine. Nevertheless, the times $\underline{t}$ and $\overline{t}$ do increase over the iterations because we collect more and more scenarios which slows down the probability evaluations in each optimization. The time $\overline{t}$ to calculate the iterate $\overline{x}_k$ is quite small since it does not need to evaluate probabilities which is numerically costly in these examples.

Next to the extraction profiles, we can also consider the corresponding T-discretizations $T_k \subseteq T$ that increase over the iterations $k \in \mathbb{N}$ as represented in Figure 6.1.

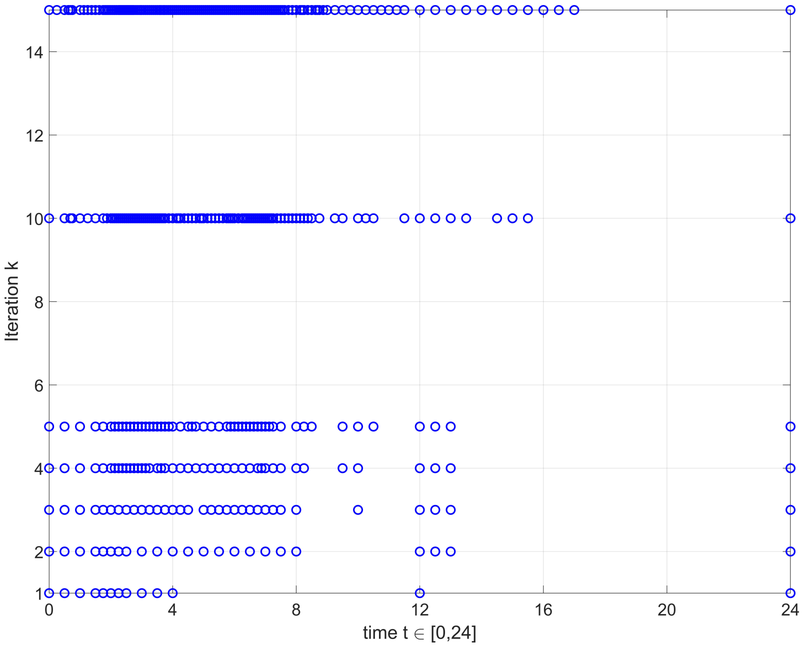


Figure 6.1: UHLAD discretization points in different iterations for price signal $\tilde{c} = (1, 0)$

Because we want to take as much water out of the reservoir as possible in the time interval $[0, 12]$, we stay close to the lower water level $\underline{l}$. Consequently, a lot of time points in this interval are critical depending on the inflow realization. Although there might be a lot of worst-case time points, Table 6.1 indicates that just a few points are necessary to calculate a rather good approximation of the optimal solution.

As we understand the worst-case time points, we can focus on the development of the sets of feasible realizations. Therefore, Figure 6.2 shows how the different sets $\Omega(\underline{x}_k, T_k)$, $\Omega(\underline{x}_k)$ and $\Omega(\overline{x}_k)$ behave in iteration 1 and 10, where the real sets of feasible realizations are approximated by a fine uniform grid over $[0, 24]$ with 10^5 points.

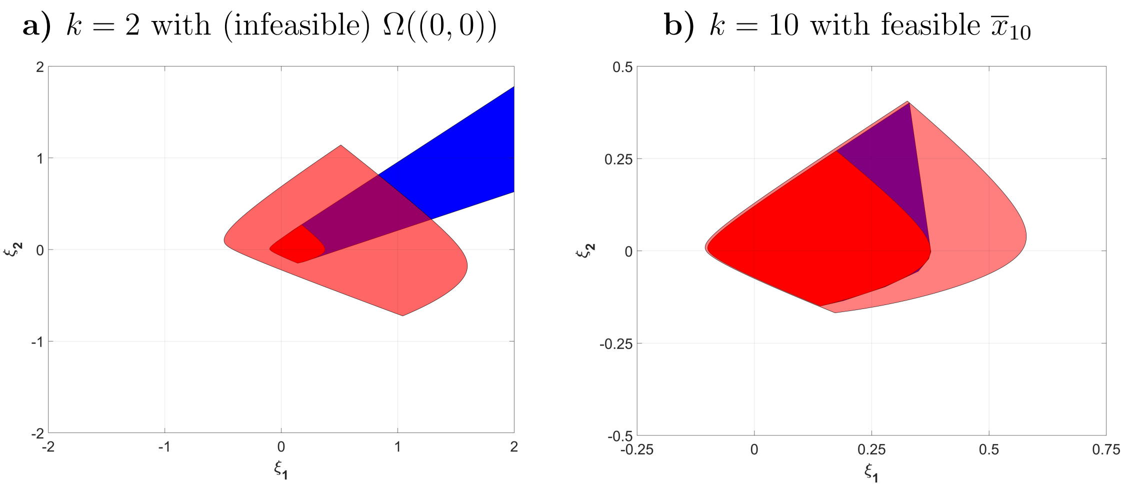


Figure 6.2: Sets of feasible realizations $\Omega(\overline{x}_k)$ (transparent red), $\Omega(\underline{x}_k, T_k)$ (blue) and $\Omega(\underline{x}_k)$ (filled red) for price signal $\tilde{c} = (1, 0)$

The comparison of the sets for the two different iterations stresses that we might need a lot of scenarios to describe the set of feasible realizations properly w.r.t. the Hausdorff-metric. This is due to the small probability of the realizations which are further away from the expected value $\tilde{\mu} = (0, 0)$ in this example.
For the next problem instance, we choose the price signal to be $\tilde{c} = (0, 1)$. We expect that an optimal extraction profile saves the water from the first time interval $[0, 12]$ to pump out more water in the second time interval. In Table 6.3, we can find the corresponding iterates and calculation times. Apparently, we do not find an inner approximation $\overline{x}_k$ in a reasonable time for this problem instance, while the outer approximation nearly converges after the first iteration.

iteration	# point in	Approximations		time in [sec]			
k	$T_k \subseteq [0, 24]$	$\underline{x}_k^*$	$\overline{x}_k^*$	t_S	$\underline{t}$	$\overline{t}$	t_{total}
1	13	$(0, 0.2009)$	infeasible	31.1	55.0	–	86.1
2	23	$(0, 0.2009)$	infeasible	761	51	–	812

Table 6.3: Numerical results of UHLAD sandwiching approach for PWR$_1$ instance with $\tilde{c} = (0, 1)$

The corresponding T-discretization is represented in Figure 6.3. Considering the two iterations with their discretization points, there is just a small difference as the time points around $t = 12$ and $t = 24$ seem to be important.

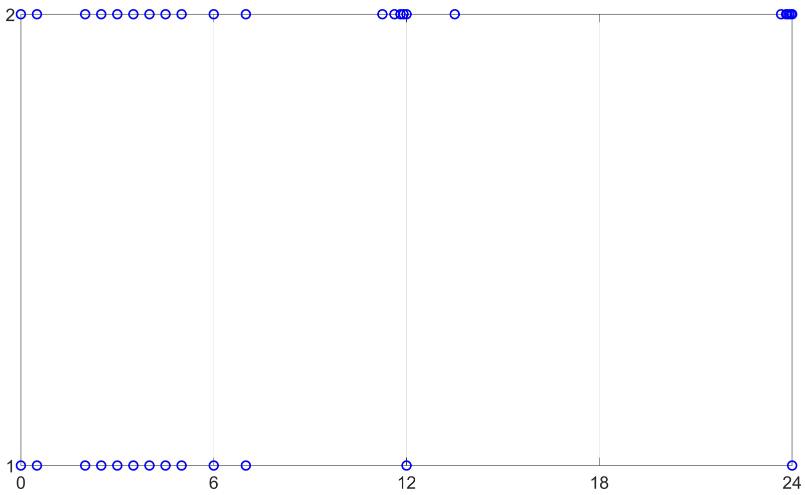


Figure 6.3: UHLAD discretization points in different iterations for price signal $\tilde{c} = (0, 1)$

The behavior of the discretization points in combination with the missing inner approximations can be explained by considering the sets of feasible realizations in Figure 6.4.

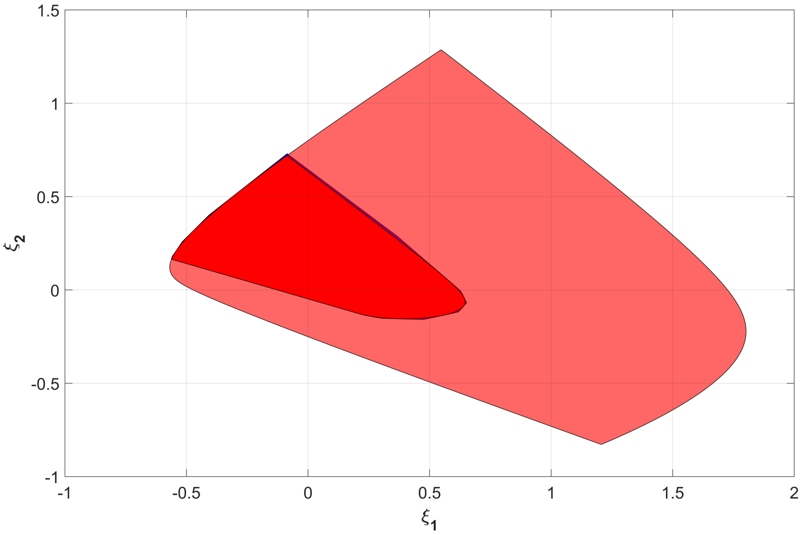


Figure 6.4: Sets of feasible realizations $\Omega(\overline{x}_2)$ (transparent red), $\Omega(\underline{x}_2, T_2)$ (blue) and $\Omega(\underline{x}_2)$ (filled red)

With this figure, we can understand that the inner set-approximation does not find a solution because the set $\Omega(\underline{x}_2, T_2)$ slightly extends even the (biggest) set of feasible realizations $\Omega((0, 0))$ and therefore there cannot be a feasible solution for this problem. We have to increase the iterations (and running time) such that this $\Omega(\underline{x}_k, T_k)$ can be included in $\Omega((0, 0))$ for some $k \in \mathbb{N}$ before we are able to solve the inner set-approximation problem. Nevertheless, the last figure clarifies that after two iterations the approximation of $\underline{x}_2$ is geometrically quite good and we stop the procedure with $x^* \approx (0, 0.200885)$.

After understanding the behavior of sandwiching-algorithms for instances with PWR$_1$ like structure, we now consider a bigger problem instance with $n = 24, m = 10$. Furthermore, we consider the following alternating price signal

$$\tilde{c}_i = -1 - \sin\left(\frac{2\pi i}{3}\right) \cdot 2^{-\frac{i}{12}}, i = 1, ..., 24.$$

This price signal can be represented graphically as by Figure 6.5

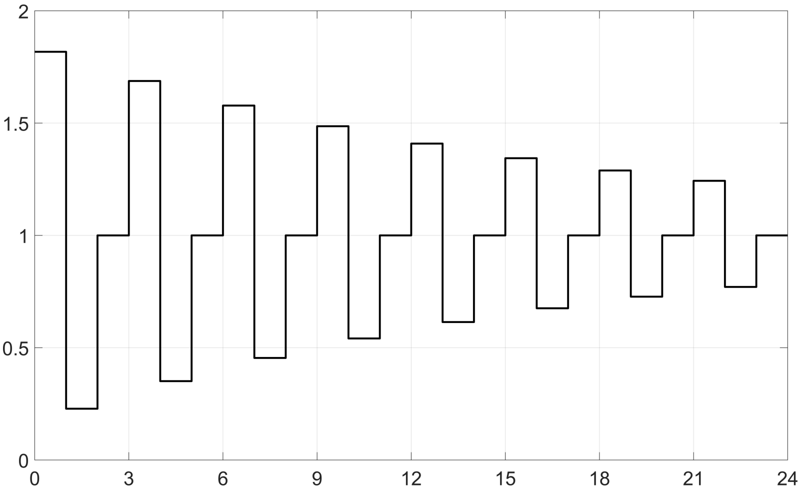


Figure 6.5: Alternating price signal $\tilde{c}$

Given the insights of the first two example problems, we expect an optimal extraction profile to store water for time periods with a low price signal and release water in timer intervals with a high price signal.
We solve this problem using the vector $x_0 = 0.05 \cdot e_{24}$, where $e_{24} = \{1, 1, ..., 1\} \in \mathbb{R}^{24}$ and the discretization $T_0 = \{0, 1, 2, ..., 24\}$. The results of the UHLAD sandwiching can be found in Table 6.4.

iteration	# point in	Approximations		time [in sec]			
k	$T_k \subseteq [0, 24]$	$f(\underline{x}_k^*)$	$f(\overline{x}_k^*)$	t_{S}	$\underline{t}$	$\overline{t}$	t_{total}
1	35	-3.36108	infeasible	56.3	1100	–	1160
5	75	-3.36100	infeasible	174	2030	–	2200

Table 6.4: Numerical results of UHLAD sandwiching for PWR$_1$ instance with $n = 24, m = 10$ and alternating cost signal $\tilde{c}$

Again, the inner set-approximation problem does not find inner approximations of the solution, but the outer approximations do not change a lot in between the first 5 iterations.

The corresponding extraction profiles $\underline{x}_k^*, k = 1, 5$ can be plotted as piece-wise constant functions over $[0, 24]$. Because the solutions $\underline{x}_1^*, \underline{x}_5^*$ differ relatively less than 0.3% (measured in $||.||_2$), plotting the single solutions is not helpful. Instead we plot just $\underline{x}_5^*$ in Figure 6.6 which represents our approximated solution for the problem.

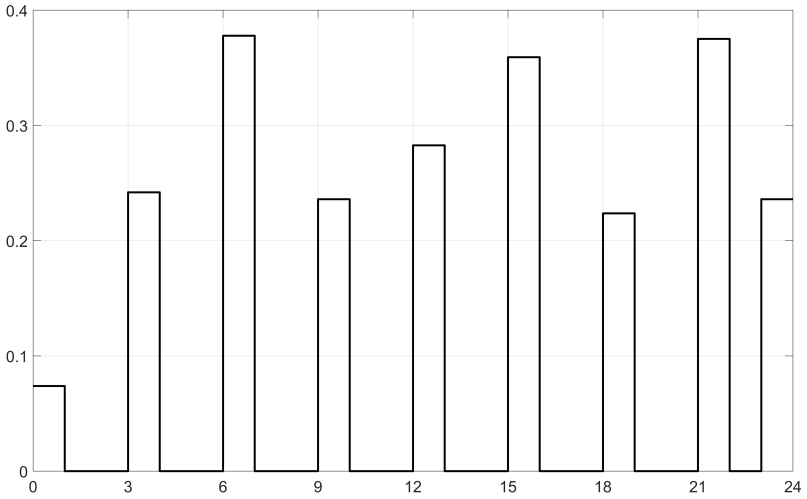


Figure 6.6: Extraction profile $\underline{x}_5^*$ for the alternating price signal

This extraction profile $\underline{x}_5^*$ adapts the alternation of the price signal. We also note that the starting discretization $T_0 = \{0, 1, ..., 24\}$ yields a good approximate $\underline{x}_1$ as it covers the main critical time points $t_{\text{crit}} \in \{1, 4, 7, 10, 13, 16, 19, 22, 24\}$. These are critical time points because at these time points we stop taking water out of the reservoir and therefore have local minima considering the filling height in many inflow realizations. Consequently, these are the time points that might violate the probust constraint most probable. Additional time points created by UHLAD do not change this solution considerably. These time points for the iterates 0, 1 and 5 are depicted in Figure 6.7.

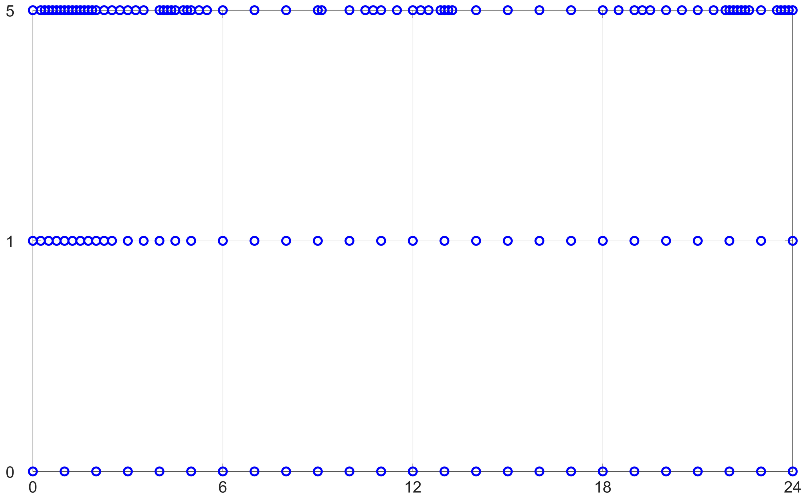


Figure 6.7: UHLAD discretization points in different iterations for alternating price signal

Next to the accumulation of new time points around the critical time points, we only generate points in the time interval $[0,3]$ because we take out water in this time according to $\underline{x}_5^*$ and therefore stay close to $\underline{l}$. Consequently, we have to consider the influence of the inflow ξ which might consume the filling height buffer $l_0 - \underline{l}$ that we start with.
As we cannot represent the set of feasible realizations in this example since it has dimension $m = 10$, we plot 100 randomly picked samples instead and check how many of the corresponding water levels lie above $\underline{l} = 0.25$ for all $t \in [0,24]$ in Figure 6.8.

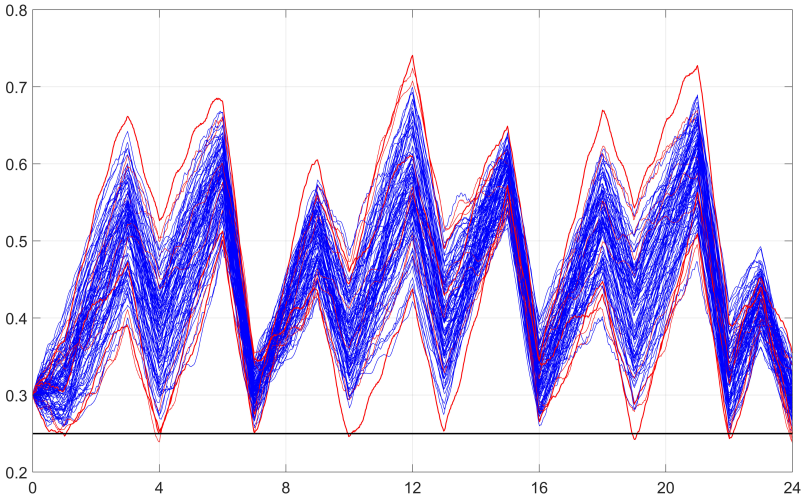


Figure 6.8: Water level over time generating 100 random inflow realizations and using extraction profile $\underline{x}_5^*$

As for 10 realizations (red lines) we can find a time point such that the filling height drops below critical water height $l_0 = 0.25$ (black line), we fulfill our condition in 90 out of 100 cases which perfectly fits our threshold $p = 0.9$. Because the inflow realizations are created randomly, we cannot guarantee this result in general, but we expect to stay close to 90% feasible inflow realizations.

Probust solution for PWR$_2$

Since we understand the numerical solution of the probust water reservoir problem instance PWR$_1$, we now consider the more complex problem instance PWR$_2$ defined by the data in (6.15) - (6.22).
We start the solving process using the UHLAD sandwiching with starting vector $x_{0,i} = 0.1$ for $i \in \{6,7,8,20,21,22,23\}$ and $x_{0,i} = 0.07$ otherwise for $i = 1,...,24$. Furthermore, we fix the starting discretization $T_0 = \{0,1,...,24\}$, a grid of size $d_1 := 1$ and a precision parameter of $\epsilon_1 = 10^{-6}$. We choose 10 additional time points in each iteration to increase the set of worst-case time points and afterwards let the optimizer take 5 steps to calculate a new extraction profile. The results can be seen in Table 6.5.

iteration	# point in	Approximations		time [in sec]			
k	$T_k \subseteq [0, 24]$	$f(\underline{x}_k^*)$	$f(\overline{x}_k^*)$	t_S	$\underline{t}$	$\overline{t}$	t_{total}
1	35	-1.885	infeasible	27.7	3330	–	3360
5	75	-1.880	infeasible	140	5150	–	5290

Table 6.5: Numerical results of UHLAD sandwiching for PWR$_2$

We also tried to compute an upper bound with the set-approximation problem based on the design $\Omega(\underline{x}_5^*, T_5)$, but the algorithm does not find a feasible solution for this problem. Because the corresponding extraction profiles $\underline{x}_k^*$ with $k = 1, 5$ do not differ a lot, we plot just $\underline{x}_5^*$ representing the (approximated) probust solution.

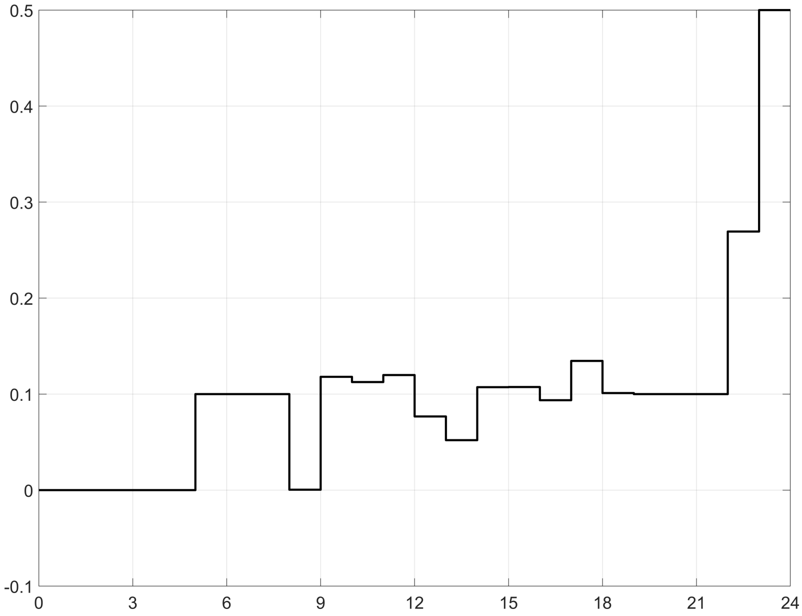


Figure 6.9: Extraction profile $\underline{x}_5^*$ over time

The development of generated discretization points over the iterations $k = 1, 2, 5$ can be seen in the Figure 6.10.

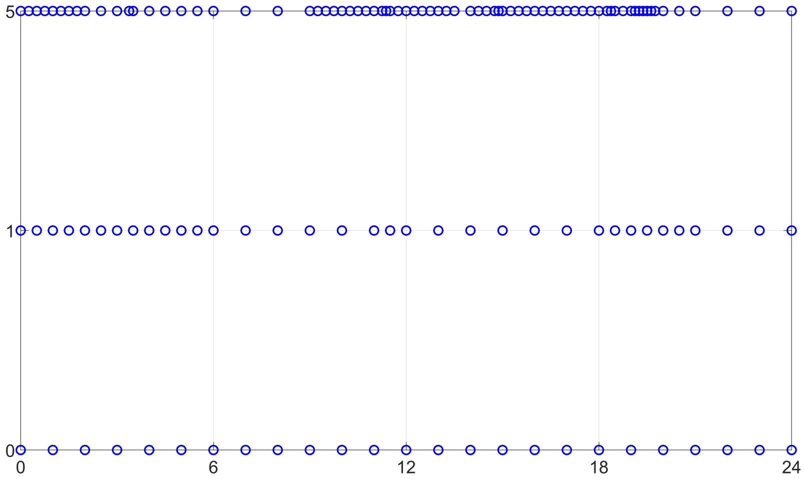


Figure 6.10: UHLAD discretization points in different iterations for PWR_2

To check if the outer approximation solution is (nearly) feasible for the problem, we generate 100 random inflows and check visually if the filling height lies in between the critical water levels in Figure 6.11.

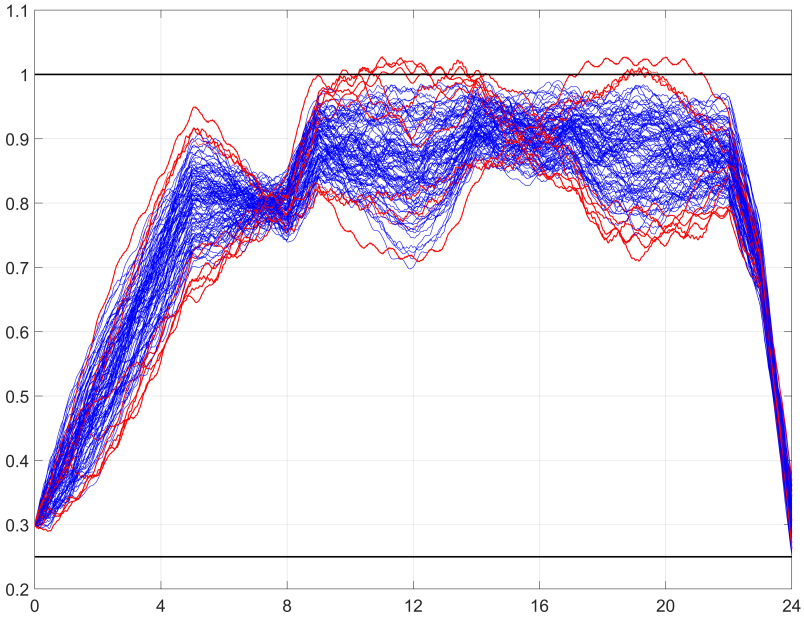


Figure 6.11: Water level over time generating 100 random inflow realizations and using extraction profile $\underline{x}_5^*$

Since 89 out of 100 scenarios respect the probust condition and $f(\underline{x}_1^*)$ and $f(\underline{x}_5^*)$ do differ just around 0.4%, we assume that $x^* \approx \underline{x}_5^*$.
Figure 6.11 illustrates that the extraction profile $\underline{x}_5^*$ stores water in the early hours of the day to work with a high pressure and in the end pumps out as much water as possible in the last time interval. This also explains the choice of critical time points in Figure 6.10 as they indicate in which time corridor we stay close to the upper water level $\bar{l}$.

Comparison with other uncertainty models

In the last sections we focused on solving the probust optimization problem instances PWR_1 and PWR_2. Now we want to compare the (approximated) probust solutions with solutions from the expected value and individual chance constrained model.
For the the expected value setting, we exchange the probability evaluation of the water level constraints by the expected value what leads to the following semi-infinite optimization problem that is linear w.r.t. $x \in X$

$$\begin{aligned}\min_{\tilde{x}\in\mathbb{R}^{24}} -\sum_{i=1}^{n} \tilde{c}_i\tilde{x}_i \text{ s.t. } \underline{l} \leq \mathbb{E}(l(\tilde{x},\tilde{\xi},t)) \; \forall t \in T,\\ -\tilde{x}_i \leq 0 \; \forall i = 1,...,24.\end{aligned}$$

In this setting we can reformulate the expected water level constraint as

$$\mathbb{E}(l(\tilde{x},\tilde{\xi},t)) = l_0 + 0.1t - \sum_{i=1}^{i^*(t)-1} \tilde{x}_i(t_i - t_{i-1}) - x_{i^*(t)}(t - t_{i^*(t)-1}).$$

Considering the individual chance constrained problem, we exchange the for-all-quantor and the probability evaluation in the water level constraint of the probust formulation to get the robust-probability problem

$$\begin{aligned}\min_{\tilde{x}\in\mathbb{R}^{24}} -\sum_{i=1}^{n} \tilde{c}_i\tilde{x}_i \text{ s.t. } &\mathbb{P}(\underline{l} \leq l(\tilde{x},\tilde{\xi},t)) \geq p \; \forall t \in T,\\ &\mathbb{P}(l(\tilde{x},\tilde{\xi},t) \leq \bar{l}) \geq p \; \forall t \in T,\\ &-\tilde{x}_i \leq 0 \; \forall i = 1,...,24.\end{aligned}$$

Here we can reformulate the water level constraint using

$$\begin{aligned}w(t) &= -(\sin(\omega_j t))_{j=1,...,m},\\ v(\tilde{x},t) &= \underline{l} - l_0 - M(t) + \sum_{i=1}^{i^*(t)-1} x_i(t_i - t_{i-1}) + x_{i^*(t)}(t - t_{i^*(t)-1})\end{aligned}$$

for an arbitrary $t \in [0, 24]$ and with $\langle\cdot,\cdot\rangle$ as the scalar product on $\mathbb{R}^m$ as

$$\mathbb{P}(\underline{l} \leq l(\tilde{x},\tilde{\xi},t)) = \mathbb{P}\left(\frac{\langle w(t),\tilde{\xi}\rangle}{\sqrt{\langle w(t),\Sigma w(t)\rangle}} \leq \frac{-v(\tilde{x},t)}{\sqrt{\langle w(t),\Sigma w(t)\rangle}}\right) = F\left(\frac{-v(x,t)}{\sqrt{\langle w(t),\Sigma w(t)\rangle}}\right) \geq p.$$

Because F as the cumulative density function of a standard normal distribution is monotonically increasing, we can invert this function and reformulate the lower water level constraint for fixed $t \in [0,T]$ as:

$$\begin{aligned}&\mathbb{P}(\underline{l} \leq l(x,\xi,t)) \geq p\\ \Leftrightarrow &\frac{-v(x,t)}{\sqrt{\langle w(t),\Sigma w(t)\rangle}} \geq F^{-1}(p)\\ \Leftrightarrow &\; g_1(x,t) := F^{-1}(p)\sqrt{\langle w(t),\Sigma w(t)\rangle} + v(x,t) \leq 0\end{aligned}$$

With the same arguments and with $l(x,\xi,t)-\bar{l} = -(\underline{l}-l(x,\xi,t))+\underline{l}-\bar{l}$, we can represent the upper water level constraint for fixed $t \in [0,T]$ as:

$$\begin{aligned}&\mathbb{P}(l(x,\xi,t) \leq \bar{l}) \geq p\\ \Leftrightarrow\ & g_2(x,t) := -F^{-1}(1-p)\sqrt{\langle w(t), \Sigma w(t)\rangle} - v(x,t) + \underline{l} - \bar{l} \leq 0\end{aligned}$$

We choose the adaptive discretization scheme from Blankenship and Falk to solve the SIP-reformulations, where we search for one discretization point in each iteration in each time interval $[t_{i-1}, t_i] = [i-1, i]$ with $i = 1, ..., 24$. The trade-off between objective value, feasibility and running time of the different uncertainty models is shown in Table 6.6.

Model	objective value f^*	feasible scenarios	solving time t [in sec]
expected	-3.483	0	12.1
robubilistic	-3.404	52	13.8
probust	-3.361	90	2200
expected	-2.032	0	13.4
robubilistic	-1.940	40	18.2
probust	-1.880	90	5290

Table 6.6: Numerical results using different uncertainty models for PWR$_1$ and PWR$_2$

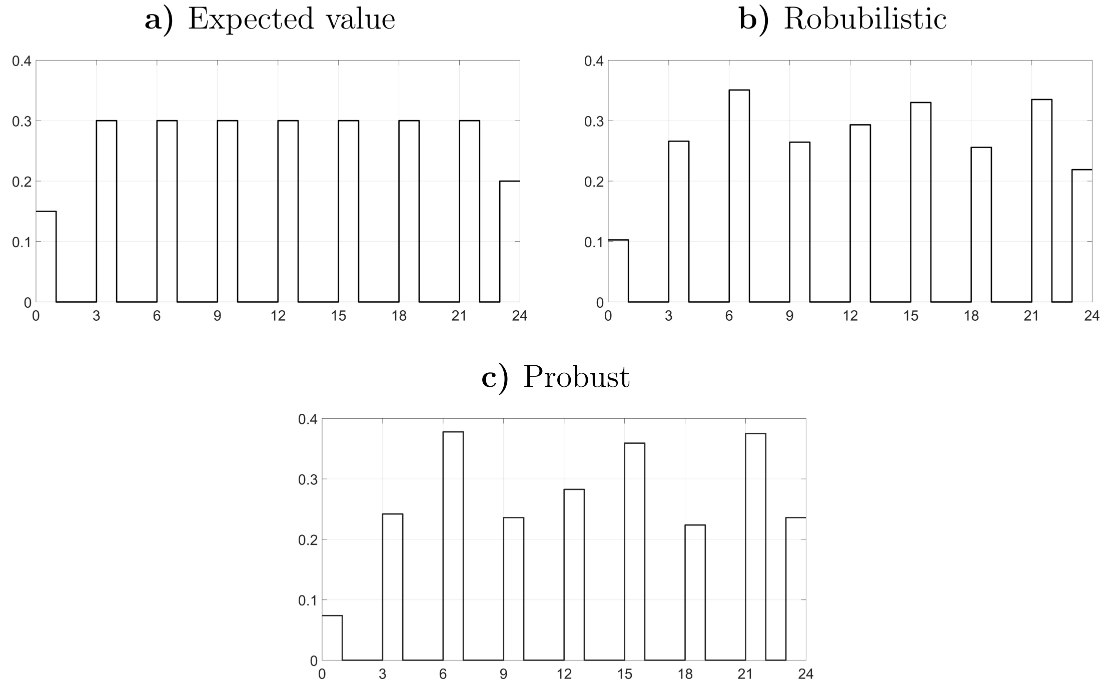
As induced by the different approaches, the expected value approach leads to the lowest objective values, while the probust extraction profile ensures feasibility for many inflow realizations. The robubilistic solution can be interpreted as a trade-off between these two aspects.

Focusing on PWR$_1$ in Table 6.6, we loose less than 4% of the objective value that corresponds to the expected value model to be able to resist 90% of uncertain inflow realizations instead of 0%, if we use the probust extraction profile. This is the strong side of the probust approach.

In contrast, the time needed to compute the probust extraction profile is more than 100 times higher than the corresponding computational times of the other two extraction profiles, although we already used modifications to make it run fast. This is due to the fact, that the probust optimization problem does not allow the reformulation of the stochastic constraint as a semi-infinite constraint that can be solved without probability evaluation. If we can allow ourselves to wait for the probust extraction profile, we would choose the probust extraction profile in terms of robustness. Otherwise, the robubilistic solution seems most attractive.

According to the problem instance PWR$_2$, we see the same behavior of the solution corresponding objective values, feasible scenarios and solving time. The only difference is that the objective values differ by 8% and the solution time for the probust extraction profile is nearly 300 times bigger than the other two. This leads to the consequence that the robubilistic solution seems even more attractive as the trade-off solution between computational time and safety while guaranteeing a low objective value.

Besides the numerical performance, we can compare the behavior of the different optimal extraction profiles. Therefore, we plot these profiles in Figure 6.12 and in Figure 6.13.

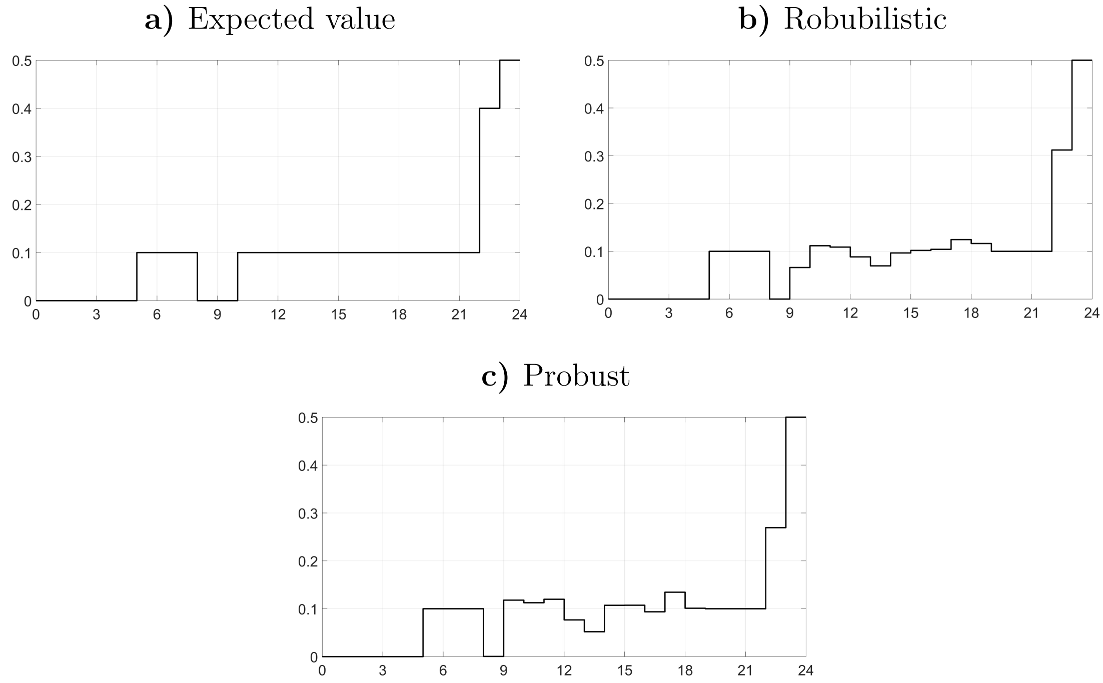


a) Expected value

b) Robubilistic

c) Probust

Figure 6.12: Optimal extraction profiles for PWR_1 of different uncertainty models



a) Expected value

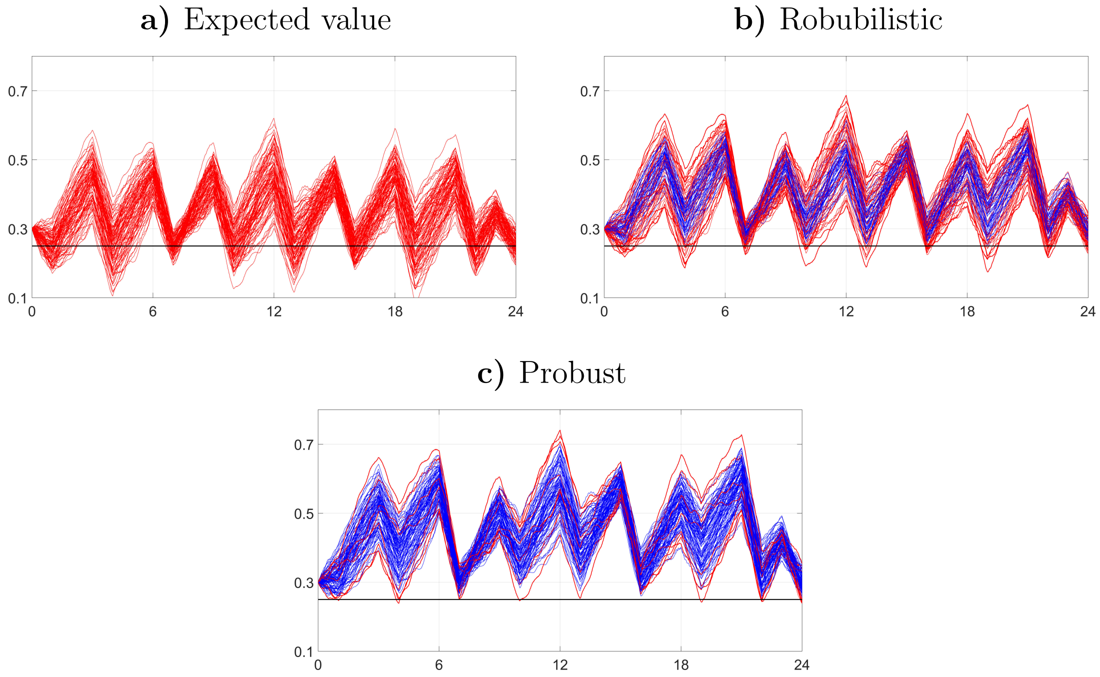
b) Robubilistic

c) Probust

Figure 6.13: Optimal extraction profiles for PWR_2 of different uncertainty models

We note that the extraction profiles corresponding to the expected value model seem calmer compared to the one related to the probust model. This is due to the fact that the probust model is more sensitive which realization are allowed to be infeasible and which do not. Therefore, it has to somehow adapt the structure of “90% of possible inflow realizations”. Concerning PWR_2 this means that when the water level $l(x, \xi, t)$ is close to the upper water level $\bar{l}$, the decision x has to compensate an increasing inflow in a certain time interval by increasing the outflow. Therefore, we would expect an even more alternating extraction profile if we would increase the number of considered time intervals $n \in \mathbb{N}$.

Furthermore, we check the robustness of the different extraction profiles by considering 100 random inflow realizations that are fixed for all optimal extraction profiles. With these, we check if the corresponding water levels respect the lower (and upper) water level constraints over time or not. Feasible realizations are plotted as blue lines, while infeasible realizations are plotted as red lines in Figure 6.14 and Figure 6.15.

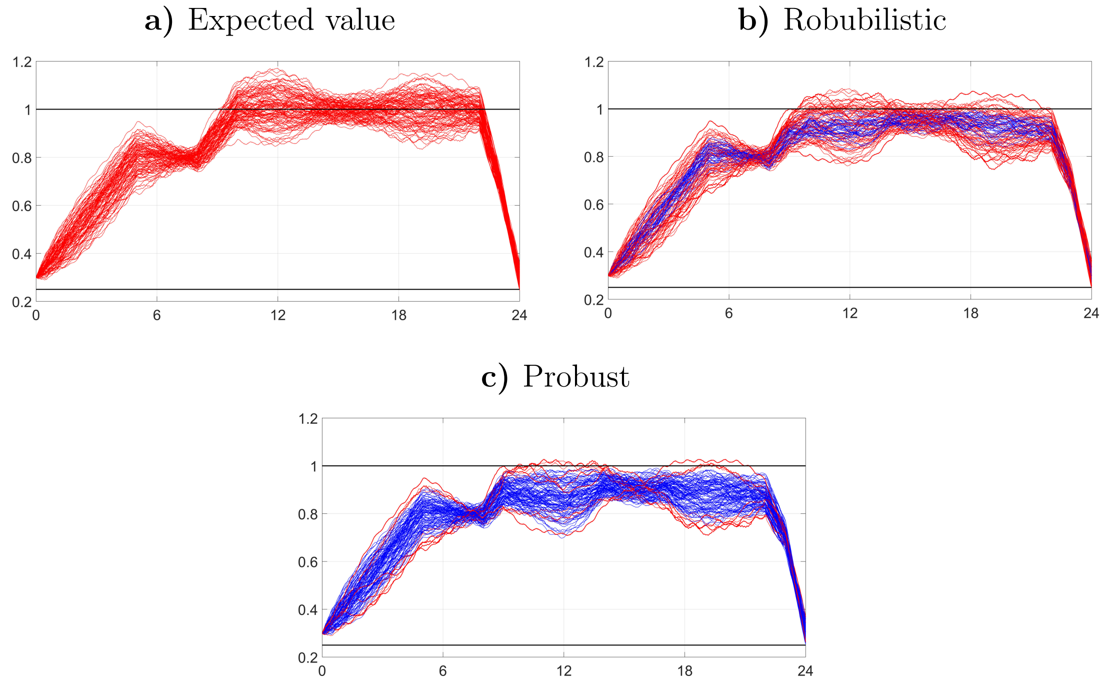

a) Expected value

b) Robubilistic

c) Probust

Figure 6.14: Feasibility check of optimal extraction profiles of PWR_1 by simulating 100 inflow realizations

These figures show us that the solution corresponding to the expected value approach is highly infeasible over all time points $[0, 24]$. Since the expected value is one fixed inflow realization, the optimal extraction profile stays at the upper or lower water level as long as possible to attain a low objective value. Because the randomly created inflows do alternate around the expected value, they are infeasible with respect to the water level constraint. The robubilistic extraction profile is more sensitive concerning the different inflow scenarios than the expected value profile. Therefore, it ensures a buffer such that



a) Expected value

b) Robubilistic

c) Probust

Figure 6.15: Feasibility check of optimal extraction profiles of PWR_2 by simulating 100 inflow realizations

10% of the inflow realizations are allowed to violate the water level constraints for each time point $t \in [0, 24]$. Consequently, we do not count (significantly) more than 10 lines violating the water level constraints at any time $t \in [0, 24]$. As these 10% of violating inflow realizations might change over time, the number of inflow realizations which violate the water level constraints for some $t \in [0, 24]$ is way bigger than 10.

Summary

In this chapter, we introduced water reservoir problems, specified four probust water problem instances, solved them by a new probust subset scheme that corresponds to a combination of a uniform discretization scheme with the HLAD and compared the optimal probust extraction profiles with the ones defined by other uncertainty models.
We have seen that for both problem instances PWR_1 and PWR_2 the “natural” discretized time horizon is a quite good approximation of the probust solution what justifies the usage of approaches that based on joint chance constrained optimization problems.
Nevertheless, we have also seen that the safety level of the whole process depends strongly on the uncertainty model. While models built on individual chance constrained optimization might be the compromise between safety, promised objective value and calculation time, they do clearly lose to the joint chance constrained based model guaranteeing the feasibility of the extraction profiles.

7 Distillation processes

In the last chapter we considered water reservoir problems. These problems are based on an inner function that is barely described by an inequality constraint which was assumed to be affine-linear with respect to decisions and realizations. Consequently, we just had to think about how to determine good scenarios to solve the problems.

In this chapter we consider a distillation process that is analytically more complex to handle than water reservoir problems. This process is described by preservation laws which induce equality constraints. Therefore, fixing a decision and a realization leads to implicitly defined problem parameters such as the temperature of the distillation as we will see. This implicit problem structure makes the corresponding probust optimization problem not only more complex, but also takes away the problem structure since some of the equality constraints are non-linear.

We start this chapter by an introduction to distillation problems in Section 7.1. Here we describe the equality constraints for our model of a distillation problem as well as roughly comment on the literature regarding distillation problems under uncertainty.

In Section 7.2 we then define our problem instance that is solved using numerical tests and the solution steps from Example 1.1.9 in Section 7.3.

7.1 Introduction to distillation problems

Process engineering deals with producing requested substances using methods from mechanics, chemistry and biology. One of these processes is the rectification where two substances are separate by inducing heat into a multi-stage system. If only one stage is used the separation process is called a distillation.

The process can be visualized by the following flow chart in Figure 7.1.

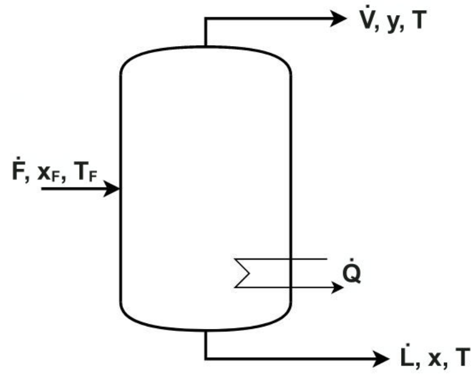


Figure 7.1: Flow chart of a continuous distillation process

To describe the distillation process mathematically, we use so called MESH-equations (see e.g. Chapter 4 in [49]) that formalize preservation laws of physical variables such as mass, pressure and energy. Rectification becomes a probust optimization problem, when rethinking the process:
Some of the parameters that are used in the MESH-equations are substance-specific and determined by experiments. We assume that these are correct up to a certain relative error which defines the set of scenarios.
Furthermore, we assume that the substances in the inflowing streams might be polluted. Therefore, we model their concentration as normal random variables with the original expected concentration as the mean value and small disturbances as their variance.
We want to maximize the volume of the end product while guaranteeing that this product has at least a given quality. This quality is given as an parameter $q \in [0,1]$ and can be measured, e.g. by the concentration of the end product $y_E \in [0,1]$. All together we can formulate the probust rectification problem of $K \in \mathbb{N}$ liquids as:

$$\mathrm{PR}_K : \min_{\chi \in X} -\dot{V} y_E \text{ s.t. } \mathbb{P}(y_E(\chi,\xi) \geq q, \mathrm{MESH}(\chi,\xi) = 0) \geq pr,$$

where $pr \in [0,1]$ is a given probability threshold, $\dot{V} \geq 0$ is the head gas stream, X is some decision space and ξ are the realizations of a random vector Z influencing the rectification.
Please note that we slightly differ the notation of the decision variable χ and the probability threshold pr to bypass a clash of notation between the process engineering notations and the mathematical notations used in this thesis so far.
The MESH-equations have the following form, where we are guided by the notation given in list of symbols and abbreviations in the beginning of this thesis:

$$\text{M: } \dot{F} x_F = \dot{L} x + \dot{V} y \tag{7.1}$$

$$\text{E: } p y_i = p_i^S(T) \gamma_i(x,T) x_i, i = 1, ..., K \tag{7.2}$$

$$\text{S: } \sum_{i=1}^{K} x_{F,i} = 1, \sum_{i=1}^{K} x_i = 1, \sum_{i=1}^{K} y_i = 1 \tag{7.3}$$

$$\text{H: } \dot{F} h_l(x_F, T_F) + \dot{Q} = \dot{L} h_l(x,T) + \dot{V} h_v(y,T) \tag{7.4}$$

Here equation M ensures the mass equilibrium meaning that substances entering the process with the feed $\dot{F}$ have to get out of the process either by exiting the head $\dot{V}$ or bottom stream $\dot{L}$.

The equation E is an extended version of Raoult's law which guarantees that the gas and liquid phase within the described stage have a constant exchange of particles. This way the corresponding substance concentrations stay unchanged over time because of the phase equilibrium. The corresponding vapor saturation pressure $p^S(T)$ is calculated as described in the appendix A.2 of Hoffmann's thesis [45] by the following approach

$$p_i^S(t) = \exp\left(c_{i,1} + \frac{c_{i,2}}{T} + c_{i,3}\log(T) + c_{i,4}T^{c_{i,5}}\right)$$

for all components $i = 1, ..., K$ with substance depending coefficients that are taken from the DIPPR Database [26].

The equation S describes that the sum of all concentrations in a stream has to add up to one.

The equation H ensures that the energy which is fed into the process by the reboiler heating and the feed temperature is the same as the energy that leaves the process in form of the temperature of the outflowing head and bottom streams. Here the enthalpy of the vapor and gas streams h_v and h_l are modeled by the linear combination of the temperature, the substance specific enthalpies $h_v(T)$ and $h_l(T)$ and the substance concentration of a stream

$$\begin{aligned} h_l(x,T) &= \sum_{i=1}^{K} x_i h_l(T), \\ h_v(y,T) &= \sum_{i=1}^{K} y_i h_v(T). \end{aligned}$$

As for the vapor saturation pressure the enthalpies h_l and h_v are defined as described in the appendix A.2 of Hoffmann's thesis [45] with coefficients from the DIPPR database [26].

Because the MESH equations describe the set of feasible realizations in PR_K, we have to extend our probust model to this case. We choose to work with implicitly given inequalities here. This way, we can represent the unknown variables used in the MESH-equations as function of the decision and the realization of the random vector, if we choose the decision space X wisely. The implicit function theorem defines this representation.

So far rectification has already been the focus of optimization under uncertainty. Not only Henrion et al. (see [42, 43]) considered in stochastic uncertainties, e.g. the feed amount, and apply different stochastic models to determine optimal controls.

But also methods from robust optimization were used to analyse the influence of empirical measured parameters like the activity coefficient within the phase equilibrium E (see Mathias [58], Burger et al. [20] or Bortz et al. [19]).

To the best of our knowledge, process optimization (or even simulation) with both kinds of uncertainties have not been published so far. This is the subject of this chapter, where we define a problem instance of a rectification problem in Section 7.2 and solve it by a set-approximation scheme that is motivated by numerical tests in Section 7.3.

7.2 Rectification problem instance

The focus of this section is to introduce a probust rectification problem instance with just one stage and two substances, a so called continuous closed single stage distillation of binary mixtures. We assume that the model for activity coefficients is correct up to a relative error of $\epsilon > 0\%$. Consequently, we can identify the possible activity coefficients with the following set of scenarios

$$\gamma \in \Gamma(\chi, \xi) \coloneqq [1-\epsilon, 1+\epsilon]^2 \cdot \gamma(\chi, \xi).$$

The reference value $\gamma(\chi, \xi)$ is calculated by a non-random-two-liquid-model (NRTL-model, see [65]).
We assume that the inflowing fluid is wine with an alcohol concentration of 12% consisting of only water and ethanol which are to be separated. To make the identification of the indices with the associates substances easier, we write $i = W$ for the water component and $i = E$ for the ethanol component of a substance stream. As the concentration of ethanol may vary using wine from different years, cultivation areas or just bottles, we model this concentration by a normal distributed random variable with expected value $\mu = 12\%$ and standard derivation $\sigma = 1\%$.
Consequently, we identify the ethanol concentration of the feed as the realizations of a random variable

$$Z \sim \mathcal{N}\left(0.12, (0.01)^2\right).$$

Moreover, we assume that we can influence the heating power $\dot{Q}$ and the pressure p to distillate in this process. These pair $(p, \dot{Q})$ defines our decision variable. This implies that we can represent the decision space as $X = [\dot{Q}_{\min}, \dot{Q}_{\max}] \times [p_{\min}, p_{\max}]$, where we chose $p_{\min} = 10^4[Pa]$ and $p_{\max} = 10^6[Pa]$ and calculate $\dot{Q}_{\min}$ and $\dot{Q}_{\max}$ by $p_{\min}$ and $p_{\max}$ as follows:
We assume that the MESH equations guarantee that no vapor stream exists. This implies $\dot{V} = 0$ for the minimal allowed heating $\dot{Q}_{\min}$. This assumption implies $\dot{L} = \dot{F}$ and $x = x_F$. Then we calculate the minimal allowed temperature $T_{\min}$ by solving the summed up pressure-equilibrium Equation (7.2) for given pressure $p_{\min}$ and the model activity coefficient $\gamma(x, T)$ by a Newton-Method. With this temperature we can use the enthalpy Equation (7.4) to calculate $\dot{Q}_{\min}$ as:

$$\begin{aligned} \dot{Q}_{\min} + \dot{F}h_l(x_F, T_F) &= \dot{L}h_l(x, T_{\min}) + \dot{V}h_v(y, T_{\min}) \\ \Leftrightarrow \dot{Q}_{\min} &= \dot{F}h_l(x_F, T_{\min}) - \dot{F}h_l(x_F, T_F) \end{aligned}$$

We can do the same with $\dot{Q}_{\max}$ by assuming that this heating implies a vanishing liquid stream ($\dot{L} = 0$) and therefore $\dot{V} = \dot{F}$ and $y = x_F$. With a maximal allowed temperature $T_{\max}$ calculated by using $p_{\max}$, this procedure leads to the representation

$$\dot{Q}_{\max} = \dot{F}h_v(x_F, T_{\max}) - \dot{F}h_l(x_F, T_F).$$

Consequently, our probust optimization problem looks like

$$\min_{\chi=(\dot{Q},p)\in X} -\dot{V}y_E \text{ s.t. } \mathbb{P}(y_E(\chi, \xi, \gamma) \geq 0.4, \text{MESH}(\chi, \xi, \gamma) = 0 \ \forall \gamma \in \Gamma(\chi, \xi)) \geq 0.9.$$

We fix the quality parameter $q \coloneqq 0.4$ such that the ethanol concentration of the gas stream should have a concentration of at least 40% independent from the activity coefficient $\gamma \in \Gamma(\chi, \xi)$ in $pr = 90\%$ of the ethanol concentration realizations in the feed x_{F_E}.
To reformulate this rectification problem as a standard probust optimization problem, we use the reference set $\hat{T} \coloneqq [1-\epsilon, 1+\epsilon]^2$ and the transformation

$$\mathcal{T}_{\hat{T}} : X \times \Xi \times \hat{T} \to \mathbb{R}^2, (\chi, \xi, z) \to \gamma(\chi, \xi) + z$$

In the next section we analyse this problem instance. We do so by focusing on the solution steps for probust optimization problems mentioned in Example 1.1.9.

7.3 Solution method and results

In this section we describe different procedures to handle the rectification problem introduced in the last section.
We use numerical simulations to understand the behavior of the worst-case activity coefficients and to guess the set of feasible realizations of the ethanol concentration in the feed. In the end, we solve the probust rectification problem for different uncertainty set parameters $\epsilon > 0$ and $\sigma > 0$.

Step $1a$: Evaluating the inner function

Inspired by the "standard" solution techniques for probust optimization problem from Example 1.1.9, we start the analysis using a fixed decision, a fixed realization of the random vector and a fixed activity coefficient and focus on evaluating the inner function.
As the inner function is defined by the quality constraint $y_E(\chi, \xi . \gamma) \geq q$, where y_E is given implicitly by the equation system $\text{MESH}(\chi, \xi, \gamma) = 0$, we have to solve the equation-system defined by the MESH-equations to be able to evaluate the quality constraint.
One way to solve these equations is by a Newton-method, where we are searching for the arguments $(\dot{L}, x, \dot{V}, y, T)$ that solve the MESH-equations. The corresponding gradients can be calculated, e.g. by central differences or differentiating the model functions and MESH-equations that were introduced in the Section 7.1.
Another method is to solve the MESH-equations by bisection. Therefore we reconsider the MESH-equations (7.1) - (7.4) and see that the equation system is linear if we know T_F, T and x.
Because we are considering a fixed feed temperature T_F and a binary substance, we can fix some $x_E \in [0,1]$, calculate $x_W = 1 - x_E$ and determine T by solving the summed up Equation (7.2)

$$\text{E'}: p = p_W^S(T)\gamma_W x_W + p_E^S(T)\gamma_E x_E$$

by a (one-dimensional) Newton-method. Using the single pressure-equilibrium equations again, we directly calculate the rate of components within the head stream by

$$y_i = \frac{p_i^S(T)}{p}\gamma_i x_i, i = W, E.$$

With these values, we calculate $\dot{V}$ and $\dot{L}$ using the first mass balance Equation (7.1) with considering the water component $i = W$ and the heat Equation (7.4).
We use the remaining mass balance equation with fixed index corresponding to the ethanol component $i = E$

$$\dot{F}x_{F,E} = \dot{L}x_E + \dot{V}y_E \tag{7.5}$$

to check if the calculated values of $(\dot{L}, x, \dot{V}, y, T)$ are correct.
Because we want to use this scheme iteratively by a bisection, we start with some liquid ethanol concentration $x_{E,0} \in [0, 1]$ and increase this concentration in the next iteration step if the left-hand side (LHS) of the feedback Equation (7.5) is bigger than the right hand side (RHS). We decrease x_E if RHS is bigger than LHS.
We stop this iterative process if the equations are fulfilled up to a precision of 10^{-6} or after a maximum of 30 iterations. This implies an absolute error of the approximated x_E less than 2^{-30}.
Because the evaluation of the inner function leads to the same values of $\dot{L}, x, \dot{V}, y, T$ up to relatives errors below 10^{-6} for both methods, we choose the bisection method in the following to evaluate the inner function because its calculation time is approximately a third of the time that the Newton method in our code needs.

Step 1*b*: Evaluating the worst-case scenario

In the next solution step, we choose activity coefficients that minimizes the ethanol concentration of the head stream

$$\min_{\gamma \in \Gamma(\chi,\xi)} y_E(\chi, \xi, \gamma).$$

Considering Equation (7.2), we represent this concentration as

$$y_E = \frac{p_E^S(T)}{p}\gamma_E x_E. \tag{7.6}$$

Using the summation Equation (7.3) for y and the phase equilibrium (7.2) for y_W, we can also try to focus on

$$1 - y_E = y_W = \frac{p_W^S(T)}{p}\gamma_W x_W. \tag{7.7}$$

As the dependencies of $p_i^S(T)x_i$ with $i = W, E$ from γ is quite complex, we fix some decision and and a realization of the random vector to solve the problem visually. Therefore, we fix the parameters $p \in \{10^4, 10^5\}, \dot{Q} \in \{5 \cdot 10^3, 10^4, 4 \cdot 10^4\}, x_{F,E} \in \{0.05, 0.12, 0.20\}$ and check 100 uniform distributed activity coefficients in the corresponding area $\Gamma(\chi, \xi)$. All examples lead to the solution $\gamma^*(\chi, \xi) = ((1+\epsilon)\gamma_W(\chi, \xi), (1-\epsilon)\gamma_E(\chi, \xi))$ as seen exemplary in Figure 7.2 that corresponds to the case $p = 10^5, \dot{Q} = 10^4, x_{F,E} = 0.12$.

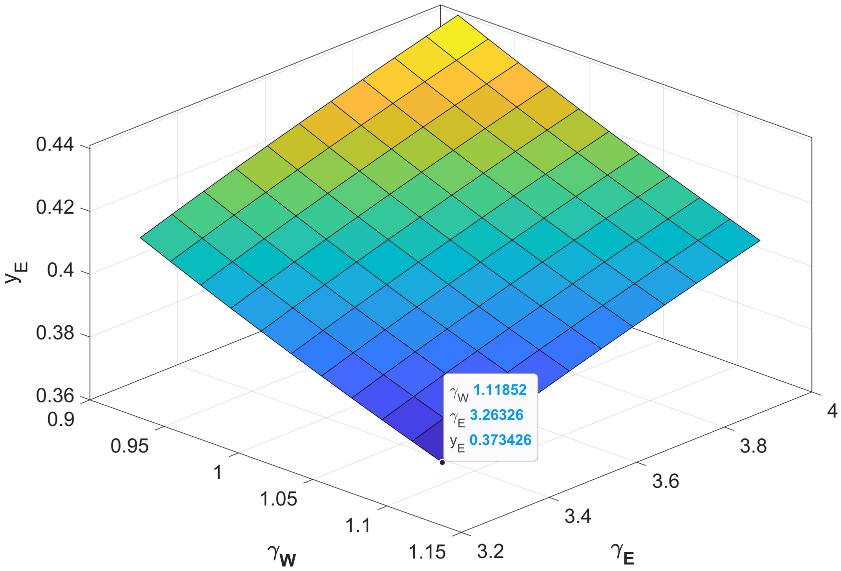


Figure 7.2: Concentration of ethanol in vapor phase y_E depending on activity coefficients γ_W, γ_E

Referring to the Equations (7.6) and (7.7) this means that the influence of γ_i seems to be stronger then the effect of $p_i^S(T)x_i$ that depend implicitly on γ for each component $i = W, E$.
Therefore, we assume in the remainder of this chapter that the worst-case activity coefficient is $\gamma^*(\chi, \xi) = (\gamma_W^*(\chi, \xi), \gamma_E^*(\chi, \xi))$, where $\gamma_W^*(\chi, \xi)$ is the maximal possible activity coefficient for water and $\gamma_E^*(\chi, \xi)$ is the minimal possible activity coefficient for ethanol within $\Gamma(\chi, \xi)$.

Step 2: Evaluating the probability

One general way to calculate the probability of

$$\mathbb{P}(y_E(\chi, \xi, \gamma^*(\chi, \xi)) \geq q)$$

is to choose a (Quasi-)Monte-Carlo simulation, where we fix a sample size of $K = 10^5$ realizations of the random vector Z, calculate the corresponding $\gamma^*(\chi, \xi)$ and check if $y_E \geq q$ after solving the MESH-equations. Because this method is extremely slow in evaluating the probability, we are interested in another evaluation strategy.

We assume that for increasing ethanol concentrations $x_{F,E}$ the ethanol concentration of the head vapor y_E does also increase. Consequently, we are interested in the critical ethanol concentration $x^*_{F,E}$ such that the corresponding ethanol concentration of the head stream y_E satisfies $y_E(\chi, x^*_{F,E}, \gamma^*(\chi, x^*_{F,E})) = q$.

Due to the assumed monotonicity between y_E and $x_{F,E}$ we can then calculate

$$\mathbb{P}(y_E(\chi, \xi, \gamma^*(\chi, \xi)) \geq q) = \mathbb{P}(x_{F,E} \geq x^*_{F,E}) = 1 - F\left(\frac{x^*_{F,E} - 0.12}{0.01}\right). \tag{7.8}$$

To calculate $x^*_{F,E}$, we solve the MESH-equations by a bisection method as explained in step $1a$, but we replace the ethanol concentration of the feed $x_{F,E}$ by an implicitly calculated variable and the value of the implicitly calculated y_E by q. Consequently, the critical ethanol feed concentration $x^*_{F,E} = x^*_{F,E}(\chi)$ depends on the inputs $\chi = (p, \dot{Q})$.

This way we just have to solve the MESH-equations once for a fixed decision χ instead of solving them for each of the 10^5 realizations in the (Quasi-)Monte-Carlo simulation.

We are allowed to assume a monotone relation between y_E and x_E as our numerical experiments show, where we fixed $p \in \{10^4, 10^5\}, \dot{Q} \in \{5 \cdot 10^3, 10^4, 4 \cdot 10^4\}$ and calculated the implicit given values of $(\dot{L}, x, \dot{V}, y, T)$ for varying $x_{F,E}$. One of these experiments with $p = 10^5[Pa], \dot{Q} = 10^4[W]$ is plotted below

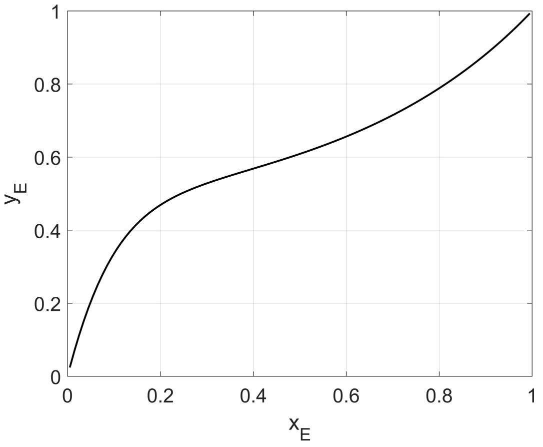


Figure 7.3: Ethanol concentration in vapor y_E depending on ethanol concentration in liquid phase x_E

Step 3: Optimizing the process

If we assume that the monotonicity assumption is fulfilled, we can reformulate the probust constraint with Equation (7.8) to a differentiable, non-linear inequality constraint

$$pr - 1 + F\left(\frac{x^*_{F,E}(\chi) - 0.12}{0.01}\right) \leq 0$$

to use gradient-based optimization tools like MATLAB's fmincon.

On the contrary, our numerical experiments support the assumption that for increasing heating $\dot{Q}$, the volume of ethanol in the head vapor stream Vy_E increases and the pureness y_E decreases. This is because more water molecules vaporize, since it is the high-boiling component.

Consequently, we can determined the optimal heating $\dot{Q}^*$ for fixed pressure as the highest feasible heating which satisfies

$$F\left(\frac{x^*_{F,E}((\dot{Q}^*, p)) - 0.12}{0.01}\right) = 1 - pr. \tag{7.9}$$

Because the heating influences y_E and therefore $x^*_{F,E}$ monotonically, we can solve Equation (7.9) for fixed pressure p by a bisection over the heating $\dot{Q}$, where we decrease the heating if the left-hand side is too low and increase it if it is too high. Alternatively, we rewrite Equation (7.9) as

$$x^*_{F,E} = 0.12 + 0.01F^{-1}(1 - pr)$$

and solve the MESH-equations given this $x^*_{F,E}$ that is independent from pressure and heating to calculate $\dot{Q}^*(p)$.

Either way this leads to an optimal heating $\dot{Q}^*(p)$ that depends on the fixed pressure $p \in [p_{\min}, p_{\max}]$. The optimization over the pressure can be performed, e.g., by a fine discretization of $[p_{\min}, p_{\max}]$. The corresponding evaluation of $\dot{V}y_E$ calculated by the MESH-equations using $p, \dot{Q}^*(p), x_{F,E} = 0.12$ and the model activity coefficients $\gamma(x, T)$ is visualized below.

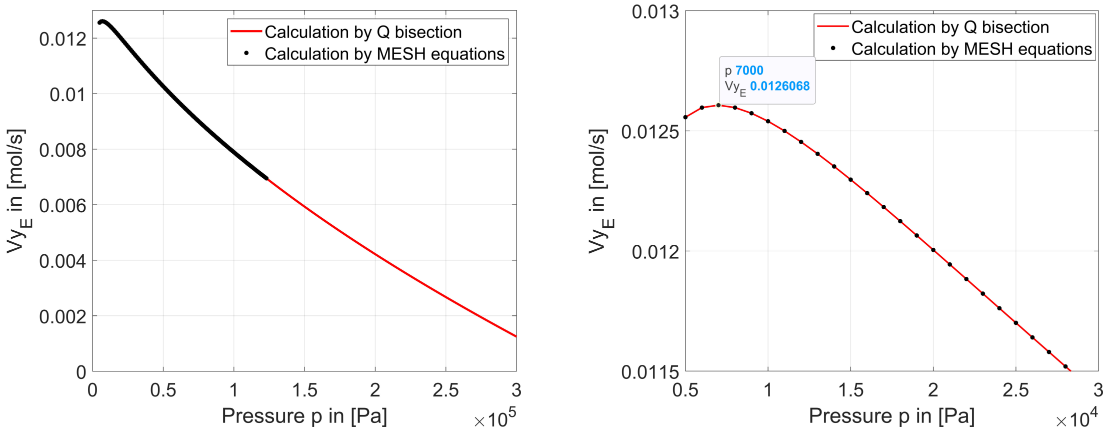


Figure 7.4: Ethanol vapor volume output over pressure (on the left) and a close-up of around its maximum (on the right)

Here the red line corresponds to the solution given by the bisection w.r.t. $\dot{Q}$, while the black dotted line corresponds to solving the MESH-equations. We see that solving the MESH-equations gets numerically unstable if the pressure is too high. Nevertheless, this approach generates sufficiently precise results for small pressures ($\leq 10^5[Pa]$) which we are interested in.

We can determine the optimal pressure graphically by $p^* \approx 7000[Pa]$, which leads to an optimal heating of $\dot{Q}^*(p^*) = 1349[W]$ and an optimal volume of $1.261 \cdot 10^{-2}[mol/s]$.

To comment shortly on running times:

- Solving the MESH-equations by a bisection with given x or y costs around 10^{-2} seconds.
- Calculating the worst-case activity coefficient costs no time as it is assumed to be known as $\gamma^*(\chi, \xi)$.
- Evaluating the probability $\mathbb{P}(y_E(\chi, \xi, \gamma^*(\chi, \xi)) \geq 0.4)$ using the monotonicity assumptions and solving the MESH-equations with given $y_E = q$ needs around 10^{-2} seconds.
- Solving for an optimal heating $\dot{Q}^*(p)$ for fixed pressure p by solving the MESH-equations for given $x^*_{F,E}, y_E$ again needs around 10^{-2} seconds. Solving for an optimal heating $\dot{Q}^*(p)$ with a bisection w.r.t. $\dot{Q}$ costs some seconds.
- Calculating the optimal operating parameters $(p, \dot{Q})$ by discretizing $[p_{\min}, p_{\max}]$ with 500 points costs some seconds or several minutes depending on how $\dot{Q}^*(p)$ is calculated.

Since we could solve the problem for a fixed relative error $\epsilon = 0.1$ and a fixed pollution $\sigma = 0.01$, we are interested in how other errors would influence the solution.
Therefore, we solve the problems which correspond to a combination of the uncertainty parameters $\epsilon \in \{10^{-8}, 0.1, 1, 10\}$ and $\sigma \in \{10^{-8}, 0.5, 1, 2\}$. The resulting optimal ethanol vapor outputs can be seen in the following figure:

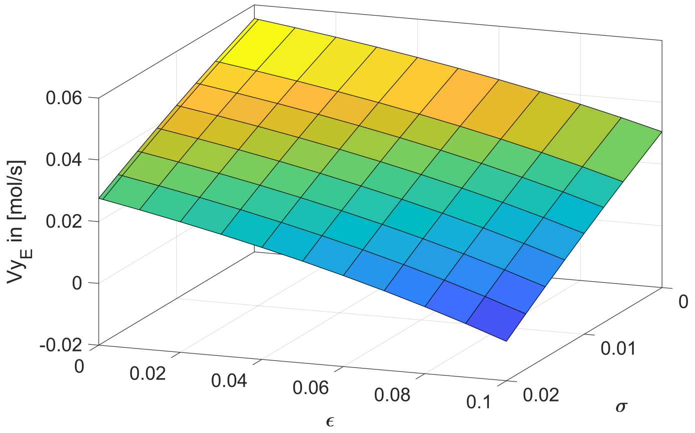


Figure 7.5: Maximum amount of ethanol in vapor Vy_E output depending on uncertainty parameters ϵ, σ

We see that smaller uncertainties ϵ, σ generate a higher objective value $\dot{V}y_E$ as one would expect. Please note that the influence of an uncertain activity coefficient ϵ is in general worse than the same uncertainty w.r.t. the ethanol concentration of the feed σ. Nevertheless, the figure shows that small errors of less than 1% do not influence the objective value too much, while bigger errors can lead to a drastical decrease.

Summary and further work

In this thesis we focused on handling probabilistic-robust (probust) optimization problems numerically. We were inspired by methods from semi-infinite optimization to handle the implied uncertainties that are an intertwined combination of uncertain parameters with and without distributional information.
We introduced useful results for (standard) probust optimization problems and related topics such as joint chance constrained optimization, semi-infinite optimization, correspondences and decision-dependent uncertainty in the first two sections of Chapter 1. Afterwards, we defined generalized probust optimization problems in Section 1.3.
Since generalized probust optimization problems define a new problem class, we gave sufficient conditions for the existence of well-defined (unique) solutions in Theorem 1.3.9. We showed that we can change the representation of a generalized probust optimization problem using appropriate transformations of the uncertainty sets without changing the feasible set or the objective function of the corresponding problem instance. In these cases, we could reduce the generalized problem to its standard variant as implied by Theorem 1.3.13.

We then started to work on numerical methods that can approximate the solution of a standard probust optimization problem in Chapters 2 and 3.
In Chapter 2, we concentrated on calculating outer approximations by discretizing the set of scenarios induced by a probust optimization problem. We were inspired by discretization schemes used to solve semi-infinite optimization problems and the analysis of probust terms to define the probust subset algorithm 3. Theorem 2.2.3 then guaranteed the convergence of the outer approximations generated by this algorithm towards a minimizer of the original probust optimization problem.
Given this convergence result, we defined some basic subset schemes and showed their convergence such as a uniform discretization scheme with increasing refinement (see Lemma 2.3.1) and two modified versions of the adaptive discretization approach from Blankenship and Falk (see Lemma 2.3.3 and Lemma 2.3.2).
With these schemes to calculate outer approximations, we focused on calculating inner approximations.
In Chapter 3 we discussed how to use a given family of measurable sets to approximate a certain probability. The resulting concept led to the inner set-approximation problem (see Definition 3.2.1) which can be interpreted as a GSIP with an additional probability evaluation constraint. The solution of such a problem defined an inner approximation for the original probust optimization problem as shown in Theorem 3.2.2.

We provided a counter example that we cannot expect a sequential set-approximation scheme to converge in a similar setting as the probust subset scheme, but we have given sufficient conditions to fix this behavior in Theorem 3.3.2.
Before we finished the chapter by applying the set-approximation approach to a probust optimization problem instance, we commented on how the structure of the inner function induces the structure of the set of feasible realizations in Proposition 3.1.7.
In Chapter 4, we discussed how to combine probust subset schemes with set-approximation schemes to define sandwiching algorithms.
While we could use both approximation methods to define upper and lower bounds for the optimal objective value separately, we were especially interested in the exchange of information between these methods. We discussed that neither a set-approximation method with information from subset schemes, nor a subset scheme with information from a set-approximation method has to lead to the convergence of the corresponding iterates in general. This is due to the structural assumptions that partly contradict each other. Nevertheless, we found a sandwiching method with exchanged information that can be solved quickly by Algorithm 7 and therefore was used in Part II of this thesis.

In Chapter 5 we introduced a generalization of design-centering problems which adjust a (design) set into another (container) set such that the design is as big as possible. We entered a random disturbance of the design or container set to the problem definition and showed that this problem then can be reformulated as a generalized probust optimization problem. As we pointed out in Section 1.3, we could find transformations to reduce these problems to standard probust optimization problems and compared the behavior of the numerical methods introduced in Chapter 2 to 4.
The main insights of this comparison were that we had to evaluate the probust function fast to solve the probust optimization problem efficiently. Moreover, structural insights of the set of feasible realizations or the behavior of worst-case scenarios were essential to understand and speed up the solving process.
With these insights we challenged probust water reservoir problems in Chapter 6. After we got a feeling how the worst-case scenarios behave, we defined a solution method that solved the probust optimization problems quite fast (compare Table 5.13 and Table 6.5). On the one hand, it is pointed out that the difference between the solution using a "natural" discretized time horizon and the probust solution is quite small (see Table 6.4). On the other hand, the comparison with other uncertainty models showed that the probust model seems appropriate in terms of safety (see Table 6.6).
In the last chapter of this thesis we considered a distillation problem to produce as much high-quality product as possible while considering randomly polluted inflow as well as a relative error in empirically measured parameters.
As this problem is defined by nonlinear equations, we had to evaluate an implicitly given inner function and did not know anything about the structure of the worst-case scenarios or the set of feasible realizations.
Based on numerical simulations, we stated assumptions how the worst-case scenarios behaves and how the set of feasible realizations is structured. With these assumptions the corresponding probust optimization problem drastically simplified and we could determine operational parameters $(p, \dot{Q})$ to run the process optimally.

In this thesis we focused on the convergence of two classes of algorithms to solve probust optimization methods. We used the freedom of defining subset and set-approximation schemes to be able to solve probust optimization problems induced by applications efficiently. During this, we had to adapt some example methods in such a way that these adapted versions react to the structure given by an application.
The whole process motivates further questions:

- In Part I of this thesis we studied the convergence of algorithms to solve probust optimization problems. One theoretical motivated question could be: What is the rate of convergence of a sandwiching algorithm given a certain probust discretization scheme? Especially the combination with a probust variant of Seidel's discretization scheme for semi-infinite problems (see [73]) seems promising.

- In Part II of this thesis we considered solely normal distributed random vectors because the spherical-radial decomposition allows to evaluate probabilities of quite complex measurable sets. One numerically motivated question could be: If the random vector is not elliptical distributed (see the book of Genz [32]), how can we evaluate the probability of a given (convex) measurable set efficiently?

- In Chapter 5 we have seen how stochastic design-centering problems can be handled by the introduced solution methods. One practically inspired question could be: Can we use stochastic design-centering problems to generate additional value, e.g. in gemstone cutting (see [72, 86])?

Bibliography

[1] W. van Ackooij and R. Henrion. “Gradient Formulae for Nonlinear Probabilistic Constraints with Gaussian and Gaussian-Like Distributions”. In: *SIAM Journal on Optimization* 24.4 (2014), pp. 1864–1889.

[2] W. van Ackooij and R. Henrion. “(Sub-)Gradient Formulae for Probability Functions of Random Inequality Systems under Gaussian Distribution”. In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 63–87.

[3] W. van Ackooij, R. Henrion, and P. Pérez-Aros. “Generalized gradients for probabilistic/robust (probust) constraints”. In: *Optimization* 69.7-8 (2019), pp. 1451–1479.

[4] W. van Ackooij et al. “Joint Chance Constrained Programming for Hydro Reservoir Management”. In: *Optimization and Engineering* 15 (2013), pp. 509–531.

[5] D. Adelhütte et al. “Joint Model of Probabilistic-Robust (Probust) Constraints Applied to Gas Network Optimization”. In: *Vietnam Journal of Mathematics* 49.4 (2020), pp. 1097–1130.

[6] C. D. Aliprantis. *Infinite Dimensional Analysis*. Vol. 317. Springer, 1999.

[7] L. Andrieu, R. Henrion, and W. Römisch. “A model for dynamic chance constraints in hydro power reservoir management”. In: *European Journal of Operational Research* 207.2 (2010), pp. 579–589.

[8] B. Basciftci, S. Ahmed, and S. Shen. “Distributionally Robust Facility Location Problem under Decision-dependent Stochastic Demand”. In: *European Journal of Operational Research* 292.2 (2021), pp. 548–561.

[9] E. M. Beale. “On Minimizing a Convex Funtion Subject to Linear Inequalities”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 17.2 (1955), pp. 173–184.

[10] G. Beer and L. Villar. “Borel Measures and Hausdorff Distance”. In: *Transaction of the American Mathematical Society* 307.2 (1988), pp. 763–772.

[11] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[12] A. Ben-Tal and A. Nemirovski. “Robust optimization - methodology and applications”. In: *Mathematical Programming* 92.3 (2002), pp. 453–480.

[13] A. Ben-Tal et al. “Adjustable robust solutions of uncertain linear programs”. In: *Mathematical Programming* 99.2 (2004), pp. 351–376.

[14] C. Berge. *Topological spaces*. Dover Books, 1963.

[15] H. Berthold et al. “On the algorithmic solution of optimization problems subject to probabilistic/robust (probust) constraints”. In: *Mathematical Methods of Operations Research* (2021), pp. 1–37.

[16] P. Billingsley and F. Topsoe. “Uniformity in weak convergence”. In: *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 7.1 (1967), pp. 1–16.

[17] J. Blankenship and J. Falk. “Infinitely constrained optimization problems”. In: *Journal of Optimization Theory and Applications* 19.2 (1976), pp. 261–281.

[18] K. C. Border. *Introduction to correspondences.* 2010. URL: `https://healy.econ.ohio-state.edu/kcb/Notes/Correspondences.pdf` (visited on 12/13/2021).

[19] M. Bortz et al. “Efficient Approach for Calculating Pareto Boundaries under Uncertainties in Chemical Process Design”. In: *Industrial and Engineering Chemistry Research* 56.44 (2017), pp. 12672–12681.

[20] J. Burger et al. “Simple Perturbation Scheme to Consider Uncertainty in Equations of State for the Use in Process Simulation”. In: *Journal of Chemical and Engineering Data* 62.1 (2016), pp. 268–274.

[21] G. Calafiore and M. Campi. “Uncertain convex programs: randomized solutions and confidence levels”. In: *Mathematical Programming* 102.1 (2004), pp. 25–46.

[22] G. D. Chakerian. “A distorted view of geometry”. In: *MAA, Mathematical Plums, Washington, DC* (1979), pp. 130–150.

[23] A. Charnes, W. Cooper, and G. Symonds. “Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil”. In: *Management Science* 4.3 (1958), pp. 235–263.

[24] G. B. Dantzig. “Linear Programming under Uncertainty”. In: *Management Science* 1.3-4 (1955), pp. 197–206.

[25] I. Déak. “Subroutines for computing normal probabilities of sets-computer experiences”. In: *Annals of Operations Research* 100 (2000), pp. 103–122.

[26] DIPPR Project 801. *Full version: Evaluated standard thermophysical property values.* 2005. URL: `https://www.aiche.org/dippr/events-products/801-database` (visited on 03/24/2022).

[27] R. Dudley. “Metric Entropy of Some Classes of Sets with Differentiable Boundaries”. In: *Journal of Approximation Theory* 10.1 (1974), pp. 227–236.

[28] J. Dupacova et al. “Stochastic programming in water resources system planning: A case study and a comparison of solution techniques”. 1986.

[29] J. Dupacova. “Optimization under exogenous and endogenous uncertainty”. 2006.

[30] M. Farshbaf-Shaker, R. Henrion, and D. Hömberg. “Properties of Chance Constraints in Infinite Dimensions with an Application to PDE Constrained Optimization”. In: *Set-Valued and Variational Analysis* 26 (2018), pp. 821–841.

[31] A. Geletu, A. Hoffmann, and P. Li. “Analytic approximation and differentiability of joint chance constraints”. In: *Optimization* 68.10 (2019), pp. 1985–2023.

[32] A. Genz and F. Bretz. *Computation of Multivariate Normal and t Probabilities.* Springer, 2009.

[33] A. L. Gibbs and F. E. Su. “On Choosing and Bounding Probability Metrics”. In: *International Statistical Review* 70.3 (2002), pp. 419–435.

[34] M. Goerigk and A. Schöbel. “Algorithm Engineering in Robust Optimization”. In: *Algorithm Engineering.* Springer International Publishing, 2016, pp. 245–279.

[35] A. Göpfert et al. *Variational methods in partially ordered spaces.* Springer Science & Business Media, 2006.

[36] C. Gotzes et al. “On the quantification of nomination feasibility in stationary gas networks with random load”. In: *Mathematical Methods of Operations Research* 84 (2016), pp. 427–457.

[37] T. G. Grandón, H. Heitsch, and R. Henrion. “A joint model of probabilistic/robust constraints for gas transport management in stationary networks”. In: *Computational Management Science* 14 (2017), pp. 443–460.

[38] T. G. Grandón. “Stochastic Optimization under Probust and Dynamic Probabilistic Constraints: with Applications to Energy Management”. PhD thesis. Humboldt-Universität zu Berlin, May 2019.

[39] S. M. Harwood and P. I. Barton. “How to solve a design centering problem”. In: *Mathematical Methods of Operations Research* 86.1 (2017), pp. 215–254.

[40] L. Hellemo, P. I. Barton, and A. Tomasgard. “Decision-dependent probabilities in stochastic programs with recourse”. In: *Computational Management Science* 15.3 (2018), pp. 369–395.

[41] R. Henrion, C. Küchler, and W. Römisch. “Scenario reduction in stochastic programming with respect to discrepancy distances”. In: *Computational Optimization and Applications* 43.1 (2007), pp. 67–93.

[42] R. Henrion and A. Möller. “Optimization of a continuous distillation process under random inflow rate”. In: *Computers and Mathematics with Applications* 45.1-3 (2003), pp. 247–262.

[43] R. Henrion et al. “Stochastic Optimization for Operating Chemical Processes under Uncertainty”. In: *Online Optimization of Large Scale Systems.* Springer Berlin Heidelberg, 2001, pp. 457–478.

[44] R. Henrion. “Structure and stability of programs with probabilistic storage level constraints”. In: *Lecture Notes in Economics and Mathematical Systems.* Vol. 513. Springer Berlin, 2002, pp. 3–20.

[45] A. Hoffmann. “Integrated simulation and optimization of distillation-based flowsheets”. PhD thesis. TU Kaiserslautern, Dec. 2016. URL: `https://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-4447608.pdf`.

[46] W. W. Hogan. “Point-to-Set Maps in Mathematical Programming”. In: *SIAM Review* 15.3 (1973), pp. 591–603.

[47] Z. Hu, L. Hong, and L. Zhang. “A smooth Monte Carlo approach to joint chance-constrained programs”. In: *IIE Transactions* 45.7 (2013), pp. 716–735.

[48] M. Inuiguchi and M. Sakawa. “Minimax regret solution to linear programming problems with an interval objective function”. In: *European Journal of Operational Research* 86 (1994), pp. 526–536.

[49] H. Z. Kister. *Distillation design.* Vol. 1. McGraw-Hill New York, 1992.

[50] D. Klatte. “Stability of stationary solutions in semi-infinite optimization via the reduction approach”. In: *Advances in optimization.* Springer, 1992, pp. 155–170.

[51] E. Kubińska. “Approximation of Carathéodory Functions and Multifunctions”. In: *Real Analysis Exchange* 30.1 (2005), p. 351.

[52] C. Leake, R. Rubinstein, and A. Shapiro. “Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method.” In: 45.8 (1994), p. 960.

[53] M. Lejeune, F. Margot, and A. de Oliveira. “Chance-Constrained Programming with Decision-Dependent Uncertainty”. In: *SSRN Electronic Journal* (2018).

[54] C. Ling et al. “A Smoothing Implicit Programming Approach for Solving a Class of Stochastic Generalized Semi-Infinite Programming Problems”. In: *Pacific Journal of Optimization* 1.1 (2005), pp. 127–145.

[55] D. Loucks and E. van Beek. *Water Resource Systems Planning and Management.* Springer International Publishing, 2017.

[56] J. Luedtke and S. Ahmed. “A Sample Approximation Approach for Optimization with Probabilistic Constraints”. In: *SIAM Journal on Optimization* 19.2 (2008), pp. 674–699.

[57] K. Margellos, P. Goulart, and J. Lygeros. “On the Road Between Robust Optimization and the Scenario Approach for Chance Constrained Optimization Problems”. In: *IEEE Transactions on Automatic Control* 59.8 (2014), pp. 2258–2263.

[58] P. M. Mathias. “Sensitivity of Process Design to Phase Equilibrium—A New Perturbation Method Based Upon the Margules Equation”. In: *Journal of Chemical & Engineering Data* 59.4 (2013), pp. 1006–1015.

[59] B. K. Pagnoncelli, S. Ahmed, and A. Shapiro. “Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications”. In: *Journal of Optimization Theory and Applications* 142.2 (2009), pp. 399–416.

[60] A. Prékopa and T. Szántai. “An optimal regulation of a storage level with application to the water level regulation of a lake”. In: *European Journal of Operational Research* 3.3 (1979), pp. 175–189.

[61] A. Prékopa. *Stochastic Programming.* Kluwer, Dordrecht, 1995.

[62] H. Rahimian and S. Mehrotra. *Distributionally Robust Optimization: A Review.* 2019. URL: `https://arxiv.org/pdf/1908.05659.pdf` (visited on 11/18/2020).

[63] R. Reemtsen and S. Görner. “Numerical Methods for Semi-Infinite Programming: A Survey”. In: *Nonconvex Optimization and Its Applications: Semi-infinite optimization.* Springer, 1998, pp. 195–275.

[64] R. Reemtsen. “Discretization methods for the solution of semi-infinite programming problems”. In: *Journal of Optimization Theory and Applications* 71.1 (1991), pp. 85–103.

[65] H. Renon and J. M. Prausnitz. “Local compositions in thermodynamic excess functions for liquid mixtures”. In: *AIChE journal* 14.1 (1968), pp. 135–144.

[66] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis.* Springer Science and Business Media, 2009.

[67] M. Rothschild and J. Stiglitz. "Increasing Risk: I. A Definition". In: *Journal of Economic Theory* 2 (1970), pp. 225–243.

[68] R. Y. Rubinstein, A. Shapiro, and S. Uryasev. "The Score Function Method". URL: `https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.24.6482&rep=rep1&type=pdf` (visited on 08/13/2021).

[69] L. Savage. "The Theory of Statistical Decision". In: *Journal of the American Statistical Association* 46.253 (1951), pp. 55–67.

[70] H. E. Scarf. "A min-max solution to an inventory problem". In: *Studies in Mathematical Theory of Inventory and Production.* Stanford University Press, 1958, pp. 201–209.

[71] J. Schwientek, T. Seidel, and K.-H. Küfer. "A transformation-based discretization method for solving general semi-infinite optimization problems". In: *Mathematical Methods of Operations Research* 93.1 (2020), pp. 83–114.

[72] J. Schwientek. "Modellierung und Lösung parametrischer Packungsprobleme mittels semi-infiniter Optimierung". PhD thesis. Technische Universität Kaiserslautern, Jan. 2013.

[73] T. Seidel and K.-H. Küfer. "An adaptive discretization method solving semi-infinite optimization problems with quadratic rate of convergence". In: *Optimization* (2020), pp. 1–29.

[74] F. Shan, L. Zhang, and X. Xiao. "A Smoothing Function Approach to Joint Chance-Constrained Programs". In: *Journal of Optimization Theory and Applications* 163.1 (2014), pp. 181–199.

[75] A. Shapiro, D. Dentcheva, and A. Ruszcynski. *Lectures on Stochastic Programming - Modeling and Theory.* Society for Industrial, Applied mathematics, and the Mathematical Programming Society, 2009.

[76] A. L. Soyster. "Convex programming with set-inclusive constraints and applications to inexact linear programming". In: *Operations Research* 21 (1973), pp. 1154–1157.

[77] O. Stein and G. Still. "On generalized semi-infinite optimization and bilevel optimization". In: *European Journal of Operational Research* 142 (2002), pp. 444–462.

[78] O. Stein. *Bi-Level Strategies in Semi-Infinite Programming.* Springer Science and Business Media, 2003.

[79] O. Stein. "A semi-infinite approach to design centering". In: *Optimization with Multivalued Mappings.* Springer US, 2006, pp. 209–228.

[80] O. Stein. "How to solve a semi-infinite optimization problem". In: *European Journal of Operations Research* 223.2 (2012), pp. 312–320.

[81] G. Still. "Generalized semi-infinite programming: Theory and methods". In: *European Journal of Operations Research* 119 (1999), pp. 301–313.

[82] G. Still. "Generalized semi-infinite programming: Numerical Aspects". In: *Optimization* 49.3 (2001), pp. 223–242.

[83] S. Uryasev. "Derivatives of probability functions and some applications". In: *Annals of Operations Research* 56.1 (1995), pp. 287–311.

[84] F. Vázquez et al. "Generalized semi-infinite programming: A tutorial". In: *Journal of Computational and Applied Mathematics* 217 (2008), pp. 394–419.

[85] E. W. Weisstein. *Steiner Inellipse - From MathWorld, A Wolfram Web Resource.* URL: `http://mathworld.wolfram.com/SteinerInellipse.html` (visited on 03/22/2022).

[86] A. Winterfeld. "Application of general semi-infinite programming to lapidary cutting problems". In: *European Journal of Operational Research* 191.3 (2008), pp. 838–854.

[87] B. Xu et al. "Stochastic Programming with a Joint Chance Constraint Model for Reservoir Refill Operation Considering Flood Risk". In: *Journal of Water Resources Planning and Management* 143.1 (2017).

[88] İ. Yanıkoğlu, B. Gorissen, and D. den Hertog. "A survey of adjustable robust optimization". In: *European Journal of Operational Research* 277.3 (2019), pp. 799–813.

[89] Y. Yuan, Z. Li, and B. Huang. "Robust optimization approximation for joint chance constrained optimization problem". In: *Journal of Global Optimization* 67.4 (2016), pp. 805–827.

Scientific career

04/2019 - 03/2022	Doctoral candidate in Mathematics Technical University of Kaiserslautern, Germany (funded by a scholarship of Fraunhofer Institute for Industrial Mathematics ITWM and the Technical University of Kaiserslautern)
10/2018 - 03/2019	Project Studies in Advanced Technologies (ProSAT) at Technical University of Kaiserslautern, Germany (funded by a scholarship of Fraunhofer Institute for Industrial Mathematics ITWM and the Technical University of Kaiserslautern)
10/2017 - 09/2018	Master studies in Economics RWTH Aachen University, Germany
10/2015 - 09/2017	Master studies in Mathematics RWTH Aachen University, Germany Master thesis: *Eine geglättete Strafiteration für optimale Steuerungsprobleme mit Nebenbedingungen*
10/2012 - 09/2015	Bachelor studies in Mathematics RWTH Aachen University, Germany Bachelor thesis: *Bestimmung der Regenbogen-verbindungszahl einiger Unicycle-Graphen*

Wissenschaftlicher Werdegang

04/2019 - 03/2022	Promotionsstudium in Mathematik Technische Universität Kaiserslautern, Deutschland (gefördert durch ein Stipendium des Fraunhofer-Instituts für Techno- und Wirtschaftsmathematik ITWM und der Technischen Universität Kaiserslautern)
10/2018 - 03/2019	Project Studies in Advanced Technologies (ProSAT) Technische Universität Kaiserslautern, Deutschland (gefördert durch ein Stipendium des Fraunhofer-Instituts für Techno- und Wirtschaftsmathematik ITWM und der Technischen Universität Kaiserslautern)
10/2017 - 09/2018	Masterstudium Wirtschaftswissenschaften RWTH Aachen, Deutschland
10/2015 - 09/2017	Masterstudium Mathematik RWTH Aachen, Deutschland Masterarbeitarbeit: *Eine geglättete Strafiteration für optimale Steuerungsprobleme mit Nebenbedingungen*
10/2012 - 09/2015	Bachelorstudium Mathematik RWTH Aachen, Deutschland Bachelorarbeit: *Bestimmung der Regenbogenverbindungszahl einiger Unicycle-Graphen*