Ruprecht-Karls-Universität Heidelberg
Fakultät für Mathematik und Informatik
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen

B. Sc. Thesis

Convex Techniques in Stochastic Linear Programming

Name: Max Jungmann
Student ID: 4155528
Supervision: Roland Herzog
Georg Müller

Date of Submission: 13.07.2023

Abstract. We consider linear programming problems where some of the problem data are subject to uncertainty. Such problems often times occur in operations research, for instance in the context of uncertain customer demands. We interpret uncertainty as randomness with a given probability distribution. This approach is called stochastic linear programming. We look for special cases where random problem data leads to (generalized) convex optimization problems. Then we develop mathematical theory as well as optimization algorithms to solve the resulting problems. We discuss convex duality theory and interior-point methods, a cutting plane method as well as a decomposition method for large-scale linear programming.

Zusammenfassung. Wir betrachten lineare Optimierungsprobleme, bei denen Teile der Eingabedaten Unsicherheiten unterliegen. Probleme dieser Art treten häufig im Bereich des Operations Researchs auf, beispielsweise im Zusammenhang mit unsicherer Kundennachfrage. Wir interpretieren Unsicherheit als Zufall, der einer gegebenen Wahrscheinlichkeitsverteilung folgt. Dieser Ansatz wird stochastische lineare Optimierung genannt. Wir untersuchen Spezialfälle, in denen zufällige Problemdaten zu (verallgemeinert) konvexen Optimierungsproblemen führen. Dann entwickeln wir die zugehörige mathematische Theorie sowie Optimierungsverfahren, um die resultierenden Probleme zu lösen. Wir diskutieren konvexe Dualitätstheorie und Innere-Punkte-Verfahren, ein Schnittebenverfahren sowie eine Zerlegungsmethode für sehr große lineare Optimierungsprobleme.

Contents

1 Introduction

One of the most important optimization models in practice is linear programming. We consider a linear program of the form

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & Tx \geq h \\ & x \geq 0 \end{aligned} \tag{LP}$$

where $A \in \mathbb{R}^{m \times n}, T \in \mathbb{R}^{s \times n}, b \in \mathbb{R}^m, h \in \mathbb{R}^s$ and $c \in \mathbb{R}^n$. A traditional approach assumes that the problem data (A, T, b, h, c) is deterministic.

However, in reality this assumption often times does not hold. If (LP) contains for instance customer demands, weather conditions, or even measurements of physical systems, they are usually subject to uncertainty. Ignoring those uncertainties can lead to solutions that turn out to be highly infeasible for the actual data and are hence not useful in practice.

Therefore, in this thesis we will consider a generalization of (LP): We still assume (A, b, c) to be deterministic but we consider

$$(T, h) := (T(\xi), h(\xi))$$

as uncertain data depending on an uncertainty vector $\xi \in \mathbb{R}^r$. The goal of this thesis is to review a way how to deal with uncertainty in optimization problems. Therefore, several approaches from the literature are discussed and related to each other.

One way to model uncertainty is to define an uncertainty set $U \subset \mathbb{R}^r$ where ξ belongs to and to require that the uncertain constraints hold for all values of ξ in this uncertainty set, i.e.

$$T(\xi)x \geq h(\xi) \quad \forall \xi \in U.$$

This is called robust programming. Of course, it is a very conservative approach. It is appropriate in situations where a violation of the uncertain constraints cannot be tolerated. These include safety critical applications. Moreover, if the underlying problem is a linear program this approach is computationally tractable for a wide range of uncertainty sets. Nevertheless, we will not pursue robust programming further in this thesis. A detailed treatment of this topic is given in Ben-Tal, Ghaoui, and Nemirovski 2009.

An alternative to model uncertainty is to assume that the problem data are random variables following a known probability distribution. We will exclusively pursue this approach here. Therefore we regard $T(\xi)$ and $h(\xi)$ as random data depending on a random vector ξ with known probability distribution.

We assume that the distribution of ξ does not depend on our decision x. A naive way to incorporate randomness is to replace the random data by their expectations $\overline{T}, \overline{h}$. This leads to the problem

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & \overline{T}x \geq \overline{h} \\ & x \geq 0. \end{aligned}$$

However, optimal solutions to this problem might violate the uncertain constraints very likely after the realization of the randomness. As an example we consider the random constraint

$$x_1 \geq \xi$$

where ξ follows a two-point distribution

$$\mathbb{P}(\xi = 10) = 0.9 = 1 - \mathbb{P}(\xi = 0).$$

Replacing ξ by its expectation leads to the constraint

$$x_1 \geq \overline{\xi} = 9. \tag{1.1}$$

If x_1 is associated with costs, it may hold $x_1 = 9$ for an optimal solution. But this means that the constraint would be violated with a 90% probability which is definitely not desirable. We rather want a statement like

$$\mathbb{P}(T(\xi)x \geq h(\xi)) \geq 0.95$$

to ensure that the constraints are satisfied with a high probability. In Chapter 2 we will consider so called single-stage models dealing with such constraints.

In some situations (e.g. in the presence of a legal commitment) even a high probabilistic guarantee may not be sufficient. However, it might be possible to take a recourse action after the realization of the randomness to compensate for a violation $h(\xi) - T(\xi)x$ of the constraint $T(\xi)x = h(\xi)$. Therefore, after a solution x is computed and a realization of the randomness took place, we solve a second stage problem

$$\begin{aligned} \text{minimize} \quad & q^\top y \quad \text{where } y \in \mathbb{R}^k \\ \text{subject to} \quad & Wy = h - Tx \\ & y \geq 0 \end{aligned}$$

with $W \in \mathbb{R}^{s \times k}, q \in \mathbb{R}^k$ to compensate for this deficiency with minimal cost. The goal is to determine x that is in a certain sense optimal also with respect to the second stage problem. Models of this type are called two-stage models and are the topic of Chapter 3. They can also be generalized to models with multiple stages. However, such multi-stage models are beyond the scope of this thesis. Further information can be found in Kall and Mayer 2010, section 3.3.

This thesis focuses mainly on the optimization parts of stochastic programming rather than the stochastic ones. In fact, we will use only basic tools from probability theory. The most important one is the multivariate normal distribution which we introduce in Section 2.2.1. Our starting point will always be the linear program (LP). Accordingly, we exclusively deal with stochastic linear programming (SLP) problems. Moreover, we restrict ourselves to convex optimization techniques. That is, we look

for special cases where SLP models lead to convex optimization problems. In particular, we leave out the very interesting field of stochastic integer programming. An excellent introduction to linear programming and convex optimization can be found in Herzog 2023, chapter 2 and 3. Methodically, Section 2.2 deals with convex duality theory and is rather theoretical. Section 2.3 and Chapter 3, however, are about algorithms (a cutting plane method and a decomposition method) to solve two specific kinds of stochastic programming problems.

2 Single-stage models

This chapter is about stochastic programming problems with only one decision stage. A decison x is made "here and now" and afterwards the realization of the randomness occurs. Unlike two-stage models, a recourse action to compensate an unadvantageous decision cannot be taken. Single-stage models are appropriate if a rare violation of the constraints is permitted. Examples include customer experience in service systems. Here it is usually sufficient if the customer experience is good in many cases but some negative customer experience can be accepted.

Single-stage models deal with constraints where the input data is not deterministic but follows a known probability distribution depending on a random vector ξ. We consider constraints of the form

$$T(\xi)x \geq h(\xi) \tag{2.1}$$

in this model type. Here $T(\xi) \in \mathbb{R}^{s\times n}$ is a random matrix and $h(\xi) \in \mathbb{R}^s$ is a random vector, both depending on the random vector $\xi \in \mathbb{R}^r$. We assume that the distribution of ξ does not depend on our decision x.

It is important to observe that the meaning of this constraints is not well defined since they contain unknown parameters. Intuitively, a vector x should be chosen such that this constraints have some "advantageous" properties. Thus, we first have to formalize the term "advantageous" in order to be able to incorporate these constraints into an optimization problem. There are many quality measures to do so (see Kall and Mayer 2010, p.73-82) and we will outline some of them below:

1. The simplest quality measure consists of taking the expectation, i.e. replacing constraint (2.1) by

   $$\mathbb{E}[T(\xi)]x \geq \mathbb{E}[h(\xi)],$$

   which results in a linear constraint. However, this is quite a poor approach as the entire information about the probability distribution is collapsed into exactly one point and the actual constraint might be violated in most cases.

2. Another approach is to bound the expected violation of the individual constraints, given that the constraints are violated, i.e.

   $$-\mathbb{E}[t_i(\xi)^\top x - h_i(\xi) \mid t_i(\xi)^\top x - h_i(\xi) < 0] \leq d_i.$$

   Here the random vector $t_i(\xi) \in \mathbb{R}^n$ denotes the i^{th} row of $T(\xi)$, $h_i(\xi) \in \mathbb{R}$ denotes the i^{th} entry of $h(\xi)$, and d_i, $i = 1, ..., s$, are some nonnegative numbers.

3. A worst case interpretation of the constraint (2.1) is given by

   $$T(\xi)x \geq h(\xi) \quad \forall \xi \in \text{supp}(\xi),$$

where $\text{supp}(\xi) \subset \mathbb{R}^r$ denotes the smallest closed set such that $\mathbb{P}(\xi \in \text{supp}(\xi)) = 1$. This approach is also called robust optimization.

4. The possibly most natural idea of dealing with random constraints is by prescribing a (high) probability level α with which the constraints have to be satisfied, i.e.

$$\mathbb{P}(T(\xi)x \geq h(\xi)) \geq \alpha.$$

Constraints of this type are called probabilistic or chance constraints. The rest of this chapter will exclusively deal with these chance constraints.

2.1 Single versus joint chance constraints

There are two different kinds of chance constraints which we will introduce and motivate in the following: Single chance constraints and joint chance constraints. Therefore, we consider the following two problems:

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \\ & \mathbb{P}(t_i(\xi)^\top x \geq h_i(\xi)) \geq \alpha_i\,,\ i = 1, ..., s, \end{aligned} \tag{SCCP}$$

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \\ & \mathbb{P}(T(\xi)x \geq h(\xi)) \geq \alpha. \end{aligned} \tag{JCCP}$$

Here $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ are deterministic. $T(\xi) \in \mathbb{R}^{s\times n}$ is a random matrix and $h(\xi) \in \mathbb{R}^s$ is a random vector, both depending on the random vector $\xi \in \mathbb{R}^r$. The random vector $t_i(\xi) \in \mathbb{R}^n$ denotes the i^{th} row of $T(\xi)$ and $h_i(\xi) \in \mathbb{R}$ denotes the i^{th} entry of $h(\xi)$. $\alpha, \alpha_i \in [0, 1]$ are given probabilities.

The first optimization problem (SCCP) is called an individual or single chance constraint problem as each probability function contains exactly one (scalar) inequality. Contrary to that, in the second problem (JCCP) all the uncertain constraints are contained in exactly one probability function. Accordingly, this is called a joint chance constraint problem.

From the modeling point of view joint chance constraints are more desirable than single chance constraints. The reason is that a joint chance constraint ensures that all the constraints are satisfied simultaneously with a given (high) probabillity level α. In contrast to that, a set of single chance constraints only ensures that each constraint individually is satisfied with a given probability α_i but makes no direct statement about the uncertain constraints as a whole. In applications one typically wants to ensure that all the constraints are satisfied with a high probability and statements about the

individual constraints are not of primary interest. However, in Section 2.2.6 we present a problem where an immediate use of single chance constraints is reasonable.

Unfortunately, from an algorithmic point of view the situation is exactly the opposite: A single chance constraint is way better to deal with algorithmically than a joint chance constraint. Nevertheless, there are two major difficulties when dealing with chance constraints.

1. In general, the evaluation of probability functions is very difficult if not impossible at all. For instance, for a continous distribution one has to compute an integral over the density function f, i.e. for $A \subset \mathbb{R}^r$
$$\mathbb{P}(A) = \int_A f(x)\, dx,$$
which is numerically intractable in high dimensions. The only available remedy is to use Monte Carlo methods which we will briefly discuss in Section 2.3.2.

2. In general, chance constraints do not define convex sets which makes optimization over them a highly problematic task. For instance, we consider a two point distribution
$$\mathbb{P}(\xi = 1) = \frac{1}{2} = \mathbb{P}(\xi = -1)$$
and the constraint
$$\mathbb{P}(\xi x \geq 1) \geq 0.5.$$
Hence, $x = 1$ and $y = -1$ are feasible for this constraint but $0 = 0.5 \cdot x + 0.5 \cdot y$ is not.

The rest of this chapter is divided in three parts. In Section 2.2 we deal with single chance constraints. Section 2.3 is about joint chance constraints where only the right hand side $h(\xi)$ is random. Section 2.4 considers joint chance constraints in full generality.

2.2 Single chance constraints

The motivation to consider single chance constraints is twofold:

1. First, there are important special cases that allow for a closed analytical expression and efficient algorithms to deal with them.

2. Second, they provide a possibility to approximate joint chance constraints as we show in Section 2.4.3.

The presentation starts with the aforementioned special case: A single chance constraint with multivariate normal distribution, i.e.
$$\mathbb{P}(t(\xi)^\top x - h(\xi) \geq 0) \geq \alpha \quad \text{with} \quad (t(\xi), h(\xi)) \sim \mathcal{N}(\mu, \Sigma). \tag{2.2}$$
To increase readability we omit the indices from (SCCP). The notation $(t(\xi), h(\xi)) \sim \mathcal{N}(\mu, \Sigma)$ means that $(t(\xi), h(\xi))$ follows a joint multivariate normal distribution.

First, we introduce the multivariate normal distribution and derive the closed form expression. Afterwards, we develop the theoretical background to solve the resulting optimization problem.

2.2.1 Multivariate normal distribution

Literature: Kall and Mayer 2010, section 2.2.3

Definition 2.1 (Multivariate normal distribution)
The random vector $\xi \in \mathbb{R}^r$ has a multivariate normal distribution if there exists a matrix $D \in \mathbb{R}^{r\times p}$ and a vector $\mu \in \mathbb{R}^r$ such that

$$\xi = D\tilde{\xi} + \mu,$$

where the components $\tilde{\xi}_i$ of the random vector $\tilde{\xi} \in \mathbb{R}^p$ are stochastically independent with standard normal distribution, i.e.

$$\tilde{\xi}_i \sim \mathcal{N}(0,1) \quad i.i.d. \quad \forall i = 1, ..., p.$$

It holds:

$$\mathbb{E}[\xi] = \mu, \quad \Sigma := \mathrm{Var}(\xi) = DD^\top.$$

If Σ is positive definite (or equivalently D has full row rank), then the multivariate normal distribution is called nondegenerate and has the density function (see Figure 2.1) $f : \mathbb{R}^r \to \mathbb{R}$ with

$$f(y) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(\Sigma)}} \cdot e^{-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)}.$$

In the univariate case the density function of the standard normal distribution reads

$$f(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}.$$

The corresponding distribution function is denoted by Φ, i.e.

$$\Phi(x) := \mathbb{P}(\xi \leq x) \quad \text{for} \quad \xi \sim \mathcal{N}(0,1).$$

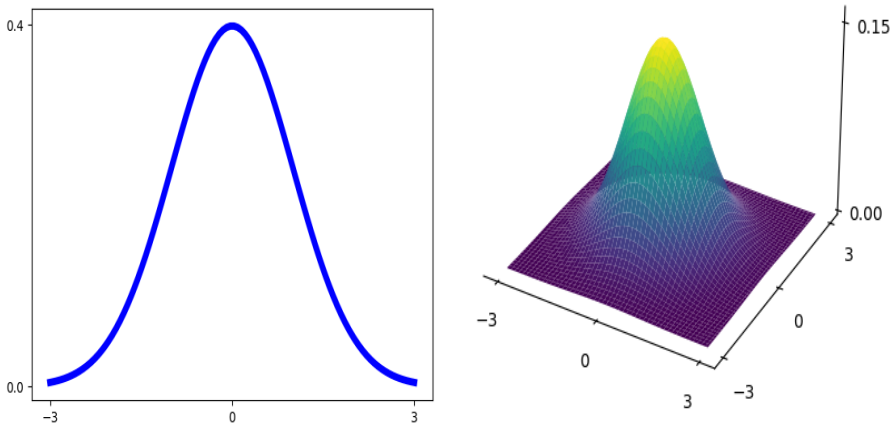

Figure 2.1: Density functions of the standard univariate and bivariate normal distribution.

We now assume the distribution of $(t(\xi), h(\xi))$ to be multivariate normal , i.e.

$$\begin{pmatrix} t(\xi) \\ h(\xi) \end{pmatrix} = \begin{pmatrix} D \\ d^\top \end{pmatrix} \tilde{\xi} + \begin{pmatrix} \mu \\ \mu_{n+1} \end{pmatrix}$$

where $D \in \mathbb{R}^{n \times p}, d \in \mathbb{R}^p, \mu \in \mathbb{R}^n, \mu_{n+1} \in \mathbb{R}$ and

$$\tilde{\xi}_i \sim \mathcal{N}(0,1) \quad \text{i.i.d.} \quad \forall i = 1, ..., p.$$

Therefore, it holds

$$\begin{aligned} t(\xi)^\top x - h(\xi) &= \xi^\top D^\top x + \mu^\top x - \xi^\top d - \mu_{n+1} \\ &= \xi^\top (D^\top x - d) + \mu^\top x - \mu_{n+1}. \end{aligned}$$

It is well known that the sum of two independent random variables with normal distribution is again normally distributed, i.e. for $\xi_1 \sim \mathcal{N}(0,1), \xi_2 \sim \mathcal{N}(0,1)$ it holds

$$(\sigma_1 \xi_1 + \mu_1) + (\sigma_2 \xi_2 + \mu_2) \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2).$$

It follows that the random variable $t(\xi)^\top x - h(\xi)$ is univariate normal distributed with

$$\mathbb{E}[t(\xi)^\top x - h(\xi)] = \mu^\top x - \mu_{n+1} \quad \text{and} \quad \operatorname{Var}(t(\xi)^\top x - h(\xi)) = \|D^\top x - d\|_2^2.$$

Assuming $\|D^\top x - d\|_2 > 0$ and applying standardization implies that

$$\frac{t(\xi)^\top x - h(\xi) - \mu^\top x + \mu_{n+1}}{\|D^\top x - d\|_2} \sim \mathcal{N}(0,1). \tag{2.3}$$

Hence, we can express the left hand side of the single chance constraint (2.2) as

$$\begin{aligned} \mathbb{P}\left(t(\xi)^\top x - h(\xi) \geq 0\right) &= 1 - \mathbb{P}\left(t(\xi)^\top x - h(\xi) \leq 0\right) \\ &= 1 - \mathbb{P}\left(\frac{t(\xi)^\top x - h(\xi) - \mu^\top x + \mu_{n+1}}{\|D^\top x - d\|_2} \leq \frac{-\mu^\top x + \mu_{n+1}}{\|D^\top x - d\|_2}\right) \\ &= 1 - \Phi\left(\frac{-\mu^\top x + \mu_{n+1}}{\|D^\top x - d\|_2}\right) \\ &= \Phi\left(\frac{\mu^\top x - \mu_{n+1}}{\|D^\top x - d\|_2}\right) \end{aligned}$$

where in the last equation we use the symmetry $\Phi(x) = 1 - \Phi(-x)$ of the standard normal distribution.

Consequently, we can write the single chance constraint (2.2) as

$$\begin{aligned} \mathbb{P}\left(t(\xi)x - h(\xi) \geq 0\right) \geq \alpha &\iff \Phi\left(\frac{\mu^\top x - \mu_{n+1}}{\|D^\top x - d\|_2}\right) \geq \alpha \\ &\iff \Phi^{-1}(\alpha)\|D^\top x - d\|_2 - \mu^\top x \leq -\mu_{n+1} \end{aligned}$$

where $\Phi^{-1}(\alpha)$ denotes the α-quantil of the standard univariate normal distribution. For $\alpha \geq 0.5$ it holds $\Phi^{-1}(\alpha) \geq 0$ and as norms are convex and the remaining terms are linear, the constraint function is a convex function and thus defines a convex set for $\alpha \geq 0.5$.

For $\|D^\top x - d\|_2 = 0$ this equivalence is also true since

$$0 \leq t(\xi)^\top x - h(\xi) = \mu^\top x - \mu_{n+1} \\ \iff -\mu^\top x \leq -\mu_{n+1}.$$

Hence, we derived the following theorem.

Theorem 2.2 (from Kall and Mayer 2010, theorem 2.5)
Let $(t(\xi), h(\xi)) \in \mathbb{R}^{n+1}$ be jointly multivariate normal distributed. Then the corresponding single chance constraint

$$\mathbb{P}\left(t(\xi)x - h(\xi) \geq 0\right) \geq \alpha \tag{2.4}$$

is equivalent to the constraint

$$\Phi^{-1}(\alpha)\|D^\top x - d\|_2 - \mu^\top x \leq -\mu_{n+1}.$$

Therefore, for $\alpha \geq 0.5$ the feasible set

$$\left\{x \;\middle|\; \mathbb{P}\left(t(\xi)^\top x - h(\xi) \geq 0\right) \geq \alpha\right\}$$

of (2.4) is convex.

Remark 2.3
The result that the value of α matters for the convexity of the single chance constraint is quite remarkable as it isn't obvious at all that the probability level should play any role for the convexity.
This result can even be strengthened. One can show that if the multivariate normal distribution is nondegenerate and the feasible set of the constraint is not $\mathbb{R}^n (n \geq 2)$, then for $\alpha < 0.5$ this constraint does not define a convex set. For a proof see Kall and Mayer 2010, theorem 2.6.

Remark 2.4
The above derivation does not only work for the normal distribution. The critical step is (2.3) where a statement about the sum of independent normally distributed random variables is made, namely that the sum is again normally distributed.
Distributions with the property that the sum of independent random variables with this distribution follows again this distribution are called stable distributions. There are only a few stable distributions with a closed form analytical expression. The most important one is definitely the univariate normal distribution. Another example ist the Cauchy distribution which allows for a very similar formulation of the corresponding single chance constraint. For details see Kall and Mayer 2010, p.106-111.

Remark 2.5
For most distributions there is no closed analytical expression of the corresponding single chance constraint. There are, however, several approximation strategies for single chance constraints. One possibility to approximate single chance constraints is presented in Ben-Tal, Ghaoui, and Nemirovski 2009, chapter 2. A good overview of approximation strategies for single chance constraints is given in Geng and Xi 2019, section 7.3.

2.2.2 Second order cone programs

We just showed that the single chance constraint

$$\mathbb{P}(t(\xi)^\top x - h(\xi) \geq 0) \geq \alpha \quad \text{with} \quad (t(\xi), h(\xi)) \sim \mathcal{N}(\mu, \Sigma)$$

is equivalent to the constraint

$$\Phi^{-1}(\alpha)\|D^\top x - d\|_2 - \mu^\top x \leq -\mu_{n+1}. \tag{2.5}$$

Constraints of this type are called second order cone constraints (SOCC). The standard second order cone is defined as

$$C := \left\{ \begin{pmatrix} x \\ t \end{pmatrix} \;\middle|\; x \in \mathbb{R}^{n-1}, t \in \mathbb{R}, \|x\|_2 \leq t \right\}.$$

It is also called the Lorentz, quadratic or ice-cream cone, see Figure 2.2. We note that C is a convex set.

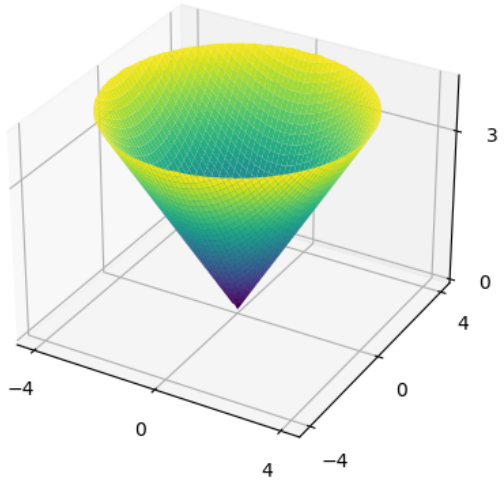

Figure 2.2: Second order cone in 3 dimensions

A general SOCC has the form

$$\|Dx + d\|_2 \leq e^\top x + f$$

where $D \in \mathbb{R}^{m_1 \times n}, d \in \mathbb{R}^{m_1}, e \in \mathbb{R}^n$ and $f \in \mathbb{R}$. This is equivalent to

$$(Dx + d, e^\top x + f) \in C.$$

Hence, the feasible set of a SOCC is convex as the inverse image of the convex cone C under an affine linear mapping. For $\Phi^{-1}(\alpha) \geq 0$ the constraint (2.5) fits into this general framework.

A second order cone program (SOCP) has the form

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & \|D_i x + d_i\|_2 \leq e_i^\top x + f_i \,, \;\; i = 1, \ldots, s. \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ and $D_i \in \mathbb{R}^{m_i\times n}, d_i \in \mathbb{R}^{m_i}, e_i \in \mathbb{R}^n$ and $f_i \in \mathbb{R}$ for $i = 1, ..., s$. Therefore, it is a convex optimization problem. We observe that the single chance constraint problem (SCCP) can be represented as a SOCP if the coefficients of the single chance constraints are all multivariate normal distributed. Setting $D_i = 0, d_i = 0, i = 1, ..., s$ shows that SOCPs contain linear programs as a special case. For an overview about SOCPs see Lobo, Vandenberghe, Boyd, and Lebret 1998.

Next, we cover the theoretical background to develop optimization algorithms for solving SOCPs. Therefore, we develop the duality theory for SOCPs. First, we derive a general statement about duality in convex optimization in Section 2.2.3. Afterwards, we apply it to the special case of SOCPs in Section 2.2.4. Finally, in Section 2.2.5 we briefly show how it can be used to build solution methods for SOCPs.

2.2.3 Lagrange duality

Literature: Geiger and Kanzow 2002, chapter 6.2

Now we introduce the concept of Lagrange duality. We consider the nonlinear program

$$\begin{array}{lll} \text{minimize} & f(x) & \text{where } x \in \mathbb{R}^n \\ \text{subject to} & g_i(x) \le 0 & \text{for } i = 1, ..., s \\ & h_j(x) = 0 & \text{for } j = 1, ..., m. \end{array} \tag{P}$$

It becomes the primal problem soon and we denote its optimal value as opt(P). Here $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^s$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are given (not necessarily convex) functions.

For $\lambda \in \mathbb{R}^s, \mu \in \mathbb{R}^m$ the corresponding Lagrange function is

$$L(x, \lambda, \mu) := f(x) + \sum_{i=1}^{s} \lambda_i g_i(x) + \sum_{j=1}^{m} \mu_j h_j(x).$$

Let x be feasible and $\lambda \ge 0$. Then it holds

$$\begin{aligned} L(x, \lambda, \mu) &= f(x) + \sum_{i=1}^{s} \underbrace{\lambda_i g_i(x)}_{\le 0} + \sum_{j=1}^{m} \underbrace{\mu_j h_j(x)}_{=0} \\ &\le f(x). \end{aligned}$$

This means that $L(x, \lambda, \mu)$ is a lower bound for $f(x)$ for every feasible x and every $\lambda \ge 0$. Hence, minimizing $L(x, \lambda, \mu)$ with respect to x for fixed μ and fixed $\lambda \ge 0$ gives a lower bound on the optimal value of P, i.e.

$$\inf_{x\in\mathbb{R}^n} L(x, \lambda, \mu) \le \text{opt(P)} \quad \forall \mu \in \mathbb{R}^m \; \forall \lambda \ge 0.$$

The Lagrange dual problem is about finding the best such lower bound. Defining

$$q(\lambda, \mu) := \inf_{x\in\mathbb{R}^n} L(x, \lambda, \mu),$$

the dual problem reads

$$\begin{aligned} &\text{maximize} && q(\lambda,\mu) \quad \text{where } \lambda \in \mathbb{R}^s,\ \mu \in \mathbb{R}^m \\ &\text{subject to} && \lambda \geq 0. \end{aligned} \tag{D}$$

We denote its optimal value by opt(D).

Hence, we derived the weak duality theorem:

Theorem 2.6 (Weak duality)
For the optimal values of the primal problem (P) and the dual problem (D) it holds

$$opt(D) \leq opt(P).$$

In linear programming the primal and dual optimal value always coincide (except for the case of primal infeasibility and dual infeasibility). This is called strong duality. However, contrary to linear programming, strong duality does not always hold as the following example shows.

Example 2.7 (from Kummer 2011, example 3.4.)
We consider the SOCP (in particular a convex optimization problem)

$$\begin{aligned} &\textit{minimize} && y \quad \textit{where } (x,y) \in \mathbb{R}^2 \\ &\textit{subject to} && \sqrt{x^2+y^2} - x \leq 0. \end{aligned} \tag{2.6}$$

$y = 0$ holds for every feasible point (x, y) . Hence, the optimal value of (2.6) is 0.

Now we consider the Lagrange dual problem to (2.6). It reads

$$\begin{aligned} &\textit{maximize} && q(\lambda) \quad \textit{where } \lambda \in \mathbb{R} \\ &\textit{subject to} && \lambda \geq 0 \end{aligned} \tag{2.7}$$

with

$$q(\lambda) := \inf_{x,y\in\mathbb{R}^2} L(x,y,\lambda) = \inf_{x,y\in\mathbb{R}^2} y + \lambda(\sqrt{x^2+y^2} - x).$$

Let $\lambda \geq 0$, y be fixed. Then we find (possibly big) x such that

$$\lambda(\sqrt{x^2+y^2} - x) < 1.$$

Hence $L(x, y, \lambda) < y + 1$. $y \to -\infty$ shows $q(\lambda) = -\infty$. This implies that also the optimal value of (2.7) is $-\infty$. Thus, strong duality does not hold in this example as $-\infty < 0$.

Remark 2.8
Lagrange duality can be seen as a generalization of linear programming duality. One can show (see Geiger and Kanzow 2002, Example 6.8) that the Lagrange dual of a linear program coincides with the well known dual linear program.

Remark 2.9
The function $q(\lambda, \mu) = \inf_x L(x, \lambda, \mu)$ is always concave. This holds as L is a linear function with respect to λ and μ and the infimum over linear functions is a concave function. Moreover it can be shown that the domain of q is convex (see Geiger and Kanzow 2002, Lemma 6.11) and hence the Lagrange dual problem is always a concave maximization problem (also if (P) was not convex!).

As shown in Example 2.7, convexity is not sufficient for strong duality to hold. But convexity and the existence of a Slater point is. We will show this in the following theorem.

Theorem 2.10 (Strong duality, from Geiger and Kanzow 2002, theorem 6.13)
Let the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., s$, from (P) be convex. Moreover let the functions $h_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1, ..., m$, be affine linear, i.e. $h_j(x) = a_j^\top x + b_j$ for some $a_j \in \mathbb{R}^n, b_j \in \mathbb{R}$.
If opt(P) is finite and there is a Slater point $\hat{x}$, i.e. a point $\hat{x}$ such that

$$g_i(\hat{x}) < 0 \ \forall i = 1, ..., s \quad \text{and} \quad h_j(\hat{x}) = 0 \ \forall j = 1, ..., m,$$

then the dual problem (D) is solvable and it holds

$$\text{opt}(D) = \text{opt}(P).$$

Proof. First we additionally assume that the vectors a_j, $j = 1, ..., m$, are linear independent.

We consider the set

$$Q := \{(y, z, w) \in \mathbb{R}^s \times \mathbb{R}^m \times \mathbb{R} \mid \exists x \in \mathbb{R}^n \quad s.t. \quad g(x) \leq y, h(x) = z, f(x) \leq w\}.$$

Using the convexity of f and g as well as the affine linearity of h one can show that Q is convex. Moreover Q is nonempty.

Next we consider the point $(0, 0, \text{opt(P)}) \in \mathbb{R}^s \times \mathbb{R}^m \times \mathbb{R}$. It holds $(0, 0, \text{opt(P)}) \notin \text{int(Q)}$. Otherwise there is $\epsilon > 0$ such that $(0, 0, \text{opt(P)} - \epsilon) \in Q$ which contradicts the definition of opt(P).

A nonempty convex set and a point that does not belong to the interior of the set can be separated by a hyperplane (see Geiger and Kanzow 2002, Lemma 2.21). Hence, there is $0 \neq (\lambda^*, \mu^*, \gamma^*) \in \mathbb{R}^s \times \mathbb{R}^m \times \mathbb{R}$, such that

$$\gamma^* \text{opt(P)} \leq (\lambda^*)^\top y + (\mu^*)^\top z + \gamma^* w \quad \forall (y, z, w) \in Q. \tag{2.8}$$

Since for $(y, z, w) \in Q$ and $\tau > 0$ also $(y, z, w + \tau) \in Q$ and opt(P) is finite, it holds $\gamma^* \geq 0$. The same argument applied to y_i yields $\lambda_i^* \geq 0$, $i = 1, ..., s$.

In the remaining part of the proof we show that indeed $\gamma^* > 0$ holds and that this implies strong duality. So at first we assume by contradiction that $\gamma^* = 0$. Then (2.8) reduces with $(y, z, w) = (g(x), h(x), f(x))$ to

$$0 \leq (\lambda^*)^\top g(x) + (\mu^*)^\top h(x) \quad \forall x \in \mathbb{R}^n. \tag{2.9}$$

In particular for the Slater point $\hat{x}$ it follows $\lambda^* = 0$ due to $\lambda^* \geq 0, g(\hat{x}) < 0$ and $h(\hat{x}) = 0$.

Using (2.9) and $h(\hat{x}) = 0$ leads to

$$0 \leq (\mu^*)^\top (h(x) - h(\hat{x})) = \left(\sum_{j=1}^{m} \mu_j^* a_j\right)^\top (x - \hat{x}) \quad \forall x \in \mathbb{R}^n.$$

Considering

$$x := \hat{x} \pm e_k$$

for a unit vector $e_k := (0, ...0, 1, 0, ...0)^\top$ leads to

$$0 \leq \pm\left(\sum_{j=1}^{m} \mu_j^* a_j\right)_k \quad \text{for } k = 1, ..., n,$$

and hence

$$\sum_{j=1}^{m} \mu_j^* a_j = 0.$$

The assumed linear independency of the a_j implies $\mu^* = 0$. Thus, $(\lambda^*, \mu^*, \gamma^*) = 0$ which is a contradiction. Hence, $\gamma^* > 0$.

W.l.o.g. we can assume $\gamma^* = 1$. Then (2.8) with $(y, z, w) = (g(x), h(x), f(x))$ leads to

$$\text{opt(P)} \leq f(x) + (\lambda^*)^\top g(x) + (\mu^*)^\top h(x) \quad \forall x \in \mathbb{R}^n. \tag{2.10}$$

$\lambda^* \geq 0$ implies

$$\text{opt(P)} \leq \inf_x L(x, \lambda^*, \mu^*) \leq \text{opt(D)}. \tag{2.11}$$

By Theorem 2.6 this inequalities have to be equalities. Hence, (λ^*, μ^*) is a optimal solution to the dual problem and opt(P) = opt(D).

Now we consider the case of linear dependent a_j , $j = 1, ..., m$. Then some equalities with indices $j \in J$ of the system

$$a_j^\top x = b_j \text{ , } j = 1, ..., m$$

can be expressed by the remaining ones. For the left hand side this holds as the a_j are linear dependent. For the right hand side this holds as $\hat{x}$ is a feasible point for this system.
Removing those redundant equalities leads to an equivalent problem where the above proof can be applied. Extending (2.10) by $\mu_j h_j$ for $j \in J$ and setting $\mu_j^* := 0$ for $j \in J$, (2.11) holds also for the original problem and the rest of the proof can be executed as above.

□

2.2.4 Duality theory for second order cone programs

Now we apply the general framework of Lagrange duality as introduced above to SOCPs.

Firstly, we derive the Lagrange dual problem to the SOCP

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & \|D_i x + d_i\|_2 \leq e_i^\top x + f_i \,, \;\; i = 1, ..., s. \end{aligned} \tag{SOCP}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $D_i \in \mathbb{R}^{m_i \times n}$, $d_i \in \mathbb{R}^{m_i}$, $e_i \in \mathbb{R}^n$ and $f_i \in \mathbb{R}$ for $i = 1, ..., s$. Secondly, we show symmetry of SOCP duality, i.e. that the dual problem of the dual problem is again the primal problem. Thirdly, we apply Theorem 2.10 to SOCPs. Fourthly, we derive optimality conditions.

Lagrange dual of SOCP

Literature: d'Aspremont 2023, p.33-37

We introduce auxiliary variables $t_i \in \mathbb{R}$, $i = 1, ..., s$, and auxiliary vectors $y_i \in \mathbb{R}^{m_i}$, $i = 1, ..., s$, and write (SOCP) as

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax - b = 0 \\ & y_i - D_i x - d_i = 0 \,, \;\; i = 1, ..., s \\ & t_i - e_i^\top x - f_i = 0 \,, \;\; i = 1, ..., s \\ & \|y_i\|_2 - t_i \leq 0 \,, \;\; i = 1, ..., s. \end{aligned} \tag{2.12}$$

Then for $\sigma \in \mathbb{R}^m$, $v_i \in \mathbb{R}^{m_i}$, $\mu_i \in \mathbb{R}$, $\lambda_i \in \mathbb{R}$, $i = 1, ..., s$, the Lagrange function reads

$$\begin{aligned} L(x, y, t, \sigma, v, \mu, \lambda) = \; & c^\top x \quad + \quad \sigma^\top (Ax - b) \quad + \quad \sum_{i=1}^{s} v_i^\top (y_i - D_i x - d_i) \\ & + \quad \sum_{i=1}^{s} \mu_i (t_i - e_i^\top x - f_i) \quad + \quad \sum_{i=1}^{s} \lambda_i (\|y_i\|_2 - t_i) \\ = \; & (c + A^\top \sigma - \sum_{i=1}^{s} (D_i^\top v_i + \mu_i e_i))^\top x \quad + \quad \sum_{i=1}^{s} (\mu_i - \lambda_i) t_i \\ & + \quad \sum_{i=1}^{s} v_i^\top y_i + \lambda_i \|y_i\|_2 \quad - \quad \sum_{i=1}^{s} (d_i^\top v_i + \mu_i f_i) \quad - \quad \sigma^\top b. \end{aligned} \tag{2.13}$$

We recall the definition of the Lagrange dual problem

$$\begin{aligned} \text{maximize} \quad & q(\sigma, v, \mu, \lambda) \quad \text{where } \sigma \in \mathbb{R}^m, v_i \in \mathbb{R}^{m_i}, i = 1, ..., s, \mu \in \mathbb{R}^s, \lambda \in \mathbb{R}^s \\ \text{subject to} \quad & \lambda \geq 0 \end{aligned}$$

with

$$q(\sigma, v, \mu, \lambda) := \inf_{x, y, t \in (\mathbb{R}^n \times \mathbb{R}^{\sum_{i=1}^{s} m_i} \times \mathbb{R}^s)} L(x, y, t, \sigma, v, \mu, \lambda).$$

We first determine the domain of $q(\sigma, v, \mu, \lambda)$, i.e. where $q(\sigma, v, \mu, \lambda) > -\infty$ holds. We see from (2.13) that

$$c + A^\top \sigma - \sum_{i=1}^{s} (D_i^\top v_i + \mu_i e_i) = 0,$$
$$\mu_i - \lambda_i = 0 \quad \forall i = 1, ..., s$$

has to hold. This is equivalent to

$$\sum_{i=1}^{s} (D_i^\top v_i + \mu_i e_i) - A^\top \sigma = c, \tag{2.15a}$$
$$\mu_i = \lambda_i \quad \forall i = 1, ..., s. \tag{2.15b}$$

Cauchy-Schwarz inequality and $\lambda \geq 0$ imply

$$\inf_{y_i} v_i^\top y_i + \lambda_i \|y_i\|_2 = \begin{cases} 0 & \text{if } \|v_i\|_2 \leq \lambda_i \\ -\infty & \text{otherwise} \end{cases}.$$

Therefore

$$\|v_i\|_2 \leq \lambda_i \quad \forall i = 1, .., s \tag{2.16}$$

has to hold.

Hence, the Lagrange dual function reads

$$q(\lambda, v, \mu, \sigma) = \begin{cases} -\sum_{i=1}^{s} (d_i^\top v_i + \mu_i f_i) - \sigma^\top b & \text{if (2.15}a\text{), (2.15}b\text{) and (2.16)} \\ -\infty & \text{otherwise} \end{cases}.$$

Accordingly, the dual problem is

$$\begin{aligned} \text{maximize} \quad & -\sum_{i=1}^{s} (d_i^\top v_i + \lambda_i f_i) - \sigma^\top b \quad \text{where } v_i \in \mathbb{R}^{m_i}, i = 1, ..., s, \lambda \in \mathbb{R}^s, \sigma \in \mathbb{R}^m \\ \text{subject to} \quad & \sum_{i=1}^{s} (D_i^\top v_i + \lambda_i e_i) - A^\top \sigma = c \\ & \|v_i\|_2 \leq \lambda_i \,, \; i = 1, ..., s \end{aligned} \tag{Dual SOCP}$$

where the constraints (2.15b) are already incorporated in the problem formulation.

Symmetry of SOCP duality

Now we show that the dual of (Dual SOCP) is equivalent to (SOCP).

One way to do this is to transform (Dual SOCP) into the shape of (SOCP). That is one has to identify the matrices and vectors $A, b, c, D_i, d_i, e_i, f_i$ in (Dual SOCP) and then can apply the above procedure. However, writing this down might be a bit confusing.

Therefore, we derive the Lagrange dual of (Dual SOCP) directly and show that it is equivalent to (SOCP). First, we write (Dual SOCP) as a minimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{s} (d_i^\top v_i + \lambda_i f_i) + \sigma^\top b \quad \text{where } v_i \in \mathbb{R}^{m_i}, i = 1, ..., s, \lambda \in \mathbb{R}^s, \sigma \in \mathbb{R}^m \\ \text{subject to} \quad & \sum_{i=1}^{s} (D_i^\top v_i + \lambda_i e_i) - A^\top \sigma - c = 0 \\ & \|v_i\|_2 - \lambda_i \leq 0\,, \ i = 1, ..., s. \end{aligned} \tag{2.17}$$

Believing in our claim we denote the dual variables with $x \in \mathbb{R}^n$ and $t \in \mathbb{R}^s$. Then the Lagrange function for (2.17) reads

$$\begin{aligned} & L(\sigma, \lambda, v, x, t) \\ &= \sum_{i=1}^{s} (d_i^\top v_i + \lambda_i f_i) + \sigma^\top b \quad + \quad x^\top \left(\sum_{i=1}^{s} (D_i^\top v_i + \lambda_i e_i) - A^\top \sigma - c \right) \quad + \quad \sum_{i=1}^{s} t_i (\|v_i\|_2 - \lambda_i) \\ &= \sigma^\top (b - Ax) \quad + \quad \sum_{i=1}^{s} \lambda_i (f_i + x^\top e_i - t_i) \quad + \quad \sum_{i=1}^{s} (d_i + D_i x)^\top v_i + t_i \|v_i\|_2 \quad - \quad c^\top x. \end{aligned} \tag{2.18}$$

Here the Lagrange dual problem reads

$$\begin{aligned} \text{maximize} \quad & q(x, t) \quad \text{where } x \in \mathbb{R}^n, \ t \in \mathbb{R}^s \\ \text{subject to} \quad & t \geq 0 \end{aligned}$$

with

$$q(x, t) := \inf_{\sigma, \lambda, v \in (\mathbb{R}^m \times \mathbb{R}^s \times \mathbb{R}^{\sum_{i=1}^{s} m_i})} L(\sigma, \lambda, v, x, t).$$

Similar as in (2.15a), (2.15b), (2.16) we obtain that $q(x, t) > -\infty$ if and only if

$$b - Ax = 0$$

$$\begin{aligned} f_i + x^\top e_i - t_i = 0 \quad & \forall i = 1, .., s \\ \|d_i + D_i x\|_2 \leq t_i \quad & \forall i = 1, .., s\,. \end{aligned}$$

Hence, the first three terms in (2.18) become zero and the Lagrange dual problem of (2.17) reads

$$\begin{aligned} \text{maximize} \quad & -c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & t_i = e_i^\top x + f_i \quad \forall i = 1, .., s \\ & \|D_i x + d_i\|_2 \leq t_i \quad \forall i = 1, .., s. \end{aligned}$$

Eliminating the variables t_i, $i = 1, ..., s$ and reversing the sign in the objective shows the equivalence to (SOCP).

Strong duality for SOCP

In order to apply Theorem 2.10 to SOCPs we introduce the notion of strict feasibility.

Definition 2.11
(SOCP) is called strictly feasible if there is a point x that is feasible for (SOCP) and satisfies

$$\|D_i x + d_i\|_2 < e_i^\top x + f_i \quad \forall i = 1, ..., s.$$

Definition 2.12
(Dual SOCP) is called strictly feasible if there is a point (σ, λ, ν) that is feasible for (Dual SOCP) and satisfies

$$\|\nu_i\|_2 < \lambda_i \quad \forall i = 1, ..., s.$$

Now we can formulate the strong duality theorem for SOCPs. It is stated and proved in Ben-Tal, Ghaoui, and Nemirovski 2009, p.455-457 in the context of conic programming. However, we use the strong duality result of Lagrange duality (Theorem 2.10) for a proof here.

Theorem 2.13 (Strong duality for SOCPs)

1. *If one of the problems (SOCP), (Dual SOCP) is strictly feasible and bounded, then the other problem is solvable, and Opt(SOCP) = Opt(Dual SOCP).*

2. *If (SOCP) and (Dual SOCP) are both strictly feasible, then both are solvable and Opt(SOCP) = Opt(Dual SOCP).*

Proof. As norms are convex functions and SOCP duality is symmetric, statement 1 follows immediately from Theorem 2.10.

In statement 2 Theorem 2.6 implies that the primal and dual problems are both bounded. Then statement 1 can be applied two times which finishes the proof. □

We observe that this is a very similar statement to linear programming duality although it is a bit weaker. In linear programming it suffices that the primal problem is feasible and bounded, or that the primal and dual problem are both feasible to ensure that the optimal values are equal and that they are also both attained. In SOCP strictly feasible points are required. The following example illustrates this:

Example 2.14 (from Alizadeh and Goldfarb 2002, p.25, 26)
We consider the SOCP

$$\begin{aligned} &\textit{minimize} && x_1 - x_2 \quad \textit{where } (x_1, x_2, x_3) \in \mathbb{R}^3 \\ &\textit{subject to} && x_3 = 1 \\ & && \left\| \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \right\|_2 \leq x_1. \end{aligned} \tag{2.20}$$

It is equivalent to

$$\begin{aligned} &\textit{minimize} && x_1 - x_2 \quad \textit{where } (x_1, x_2) \in \mathbb{R}^2 \\ &\textit{subject to} && x_1 \geq \sqrt{x_2^2 + 1}. \end{aligned}$$

The constraint ensures $x_1 - x_2 > 0$ *but* $x_1 - x_2 \to 0$ *as* $x_1 = \sqrt{x_2^2 + 1} \to \infty$ *and* $x_2 \geq 0$. *Hence, the optimal value of (2.20) is* 0 *but the problem is not solvable. Accordingly, we are in the situation of the first part of Theorem 2.13 as the optimal value of (2.20) is bounded by* 0 *and the problem is obviously strictly feasible.*

We now show that the dual optimal value is also 0 *and the dual problem is solvable. The dual problem to (2.20) is*

$$\begin{aligned} &\textit{maximize} && -\sigma \quad \textit{where } v \in \mathbb{R}^2,\ \lambda \in \mathbb{R},\ \sigma \in \mathbb{R} \\ &\textit{subject to} && \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} v + \lambda \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \sigma = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \\ & && \|v\|_2 \leq \lambda. \end{aligned}$$

The linear constraint is equivalent to $(\lambda, v) = (1, -1, \sigma)$ *and we can write the dual problem as*

$$\begin{aligned} &\textit{maximize} && -\sigma \quad \textit{where } \sigma \in \mathbb{R} \\ &\textit{subject to} && 1 \geq \sqrt{1 + \sigma^2}. \end{aligned} \tag{2.21}$$

The only feasible solution is of course $\sigma = 0$. *Hence, the optimal value of (2.21) is also* 0 *which coincides wih the primal optimal value. We note that (2.21) is not strictly feasible and the second part of Theorem 2.13 cannot be applied.*

Optimality conditions for SOCP

Now we use Theorem 2.13 to derive optimality conditions for SOCPs. Similarly to linear programming they consist of complementarity conditions.

Theorem 2.15 (SOCP optimality conditions, from Ben-Tal, Ghaoui, and Nemirovski 2009, Theorem A.2.2.)
Consider a primal-dual pair of SOCPs and let both be strictly feasible. Let x *be feasible for (SOCP) and* (σ, λ, v) *be feasible for (Dual SOCP). Then the following three statements are equivalent:*

1. x *is optimal for (SOCP) and* (σ, λ, v) *is optimal for (Dual SOCP).*

2. *The duality gap is zero, i.e.*

$$c^\top x \quad - \quad (-\sum_{i=1}^{s} (d_i^\top v_i + \lambda_i f_i) - \sigma^\top b) = 0.$$

3. *The complementarity conditions*

$$\begin{pmatrix} v_i \\ \lambda_i \end{pmatrix}^\top \cdot \begin{pmatrix} D_i x + d_i \\ e_i^\top x + f_i \end{pmatrix} = 0 \quad \forall i = 1, ..., s \tag{2.22}$$

hold.

Proof. By Theorem 2.13 we know that the optimal values of (SOCP) and (Dual SOCP) are equal, i.e. opt(SOCP) = opt(Dual SOCP). Therefore, we have the duality gap

$$c^\top x \; - \; (-\sum_{i=1}^{s}(d_i^\top v_i + \lambda_i f_i) - \sigma^\top b) \; = \; \underbrace{c^\top x - \text{opt(SOCP)}}_{a} \; + \; \underbrace{\text{opt(Dual SOCP)} - (-\sum_{i=1}^{s}(d_i^\top v_i + \lambda_i f_i) - \sigma^\top b)}_{b}$$

For any primal-dual feasible pair (x, σ, λ, v) it holds $a \geq 0,\ b \geq 0$. (x, σ, λ, v) is a pair of optimal solutions if and only if $a = 0 = b$, i.e. the duality gap is zero. This proves the equivalence of statement 1 and 2.

Now we show that statement 3 is equivalent to statement 2. As x is primal feasible it holds

$$Ax = b \quad \text{and} \quad \|D_i x + d_i\|_2 \leq e_i^\top x + f_i \;\; i = 1, ..., s.$$

The dual feasibility of (σ, λ, v) implies

$$c = \sum_{i=1}^{s}(D_i^\top v_i + \lambda_i e_i) - A^\top \sigma \quad \text{and} \quad \|v_i\|_2 \leq \lambda_i \;\; i = 1, ..., s.$$

This implies

$$\begin{aligned} c^\top x - (-\sum_{i=1}^{s}(d_i^\top v_i + \lambda_i f_i) - \sigma^\top b) &= (\sum_{i=1}^{s}(D_i^\top v_i + \lambda_i e_i) - A^\top \sigma)^\top x \quad + \quad \sum_{i=1}^{s}(d_i^\top v_i + \lambda_i f_i) + \sigma^\top b \\ &= \sigma^\top \underbrace{(-Ax + b)}_{=0} \quad + \quad \sum_{i=1}^{s} \underbrace{v_i^\top (D_i x + d_i)}_{\geq -\lambda_i (e_i^\top x + f_i)} + \lambda_i (e_i^\top x + f_i). \end{aligned}$$

Hence, the duality gap is zero if and only if

$$v_i^\top (D_i x + d_i) \; + \; \lambda_i (e_i^\top x + f_i) = 0 \quad \forall i = 1, ..., s$$

which proves statement 3.

□

Similarly as for linear programs we conclude that for a strictly feasible pair of SOCPs a point (x, σ, λ, v) is comprised of primal and dual optimal solutions if and only if

$$\begin{aligned} Ax = b, \quad & \|D_i x + d_i\|_2 \leq e_i^\top x + f_i \;\; i = 1, ..., s && \text{(primal feasibility)} \\ \sum_{i=1}^{s} (D_i^\top v_i + \lambda_i e_i) - A^\top \sigma = c, \quad & \|v_i\|_2 \leq \lambda_i \;\; i = 1, ..., s && \text{(dual feasibility)} \\ \begin{pmatrix} v_i \\ \lambda_i \end{pmatrix}^\top \cdot \begin{pmatrix} D_i x + d_i \\ e_i^\top x + f_i \end{pmatrix} = 0, \quad & i = 1, ..., s && \text{(complementarity).} \end{aligned} \tag{2.23}$$

2.2.5 Interior-point methods

Literature: Alizadeh and Goldfarb 2002, section 7.2.

The system (2.23) is the basis for so called primal-dual interior-point methods to solve SOCPs. However, the necessity of this system for optimality can only be ensured when (SOCP) and (Dual SOCP) are both strictly feasible. Therefore, all the interior-point methods for solving SOCPs assume primal-dual strict feasibility. Hence, in modeling a problem it has to be ensured that strict feasibility holds. Otherwise there might be a duality gap that lets the methods fail.

A detailed treatment of interior-point methods is far beyond the scope of this thesis. Therefore, we will only roughly sketch the ideas. A more comprehensive discussion can be found in the above mentioned paper.

We start with a definition:

Definition 2.16 (Central path)
The trajectory of points (x, σ, λ, v) *satisfying*

$$\begin{aligned} Ax = b, \quad & \|D_i x + d_i\|_2 < e_i^\top x + f_i \;\; i = 1, ..., s \\ \sum_{i=1}^{s} (D_i^\top v_i + \lambda_i e_i) - A^\top \sigma = c, \quad & \|v_i\|_2 < \lambda_i \;\; i = 1, ..., s \\ \begin{pmatrix} v_i \\ \lambda_i \end{pmatrix}^\top \cdot \begin{pmatrix} D_i x + d_i \\ e_i^\top x + f_i \end{pmatrix} = \mu, \quad & i = 1, ..., s \end{aligned} \tag{2.24}$$

for some $\mu > 0$ *is called the central path.*

We note that the above system (2.24) differs form the optimality system (2.23) in two ways:

1. The complementarity conditions

$$\begin{pmatrix} v_i \\ \lambda_i \end{pmatrix}^\top \cdot \begin{pmatrix} D_i x + d_i \\ e_i^\top x + f_i \end{pmatrix} = 0, \quad i = 1, ..., s$$

are replaced by the relaxed conditions

$$\begin{pmatrix} v_i \\ \lambda_i \end{pmatrix}^\top \cdot \begin{pmatrix} D_i x + d_i \\ e_i^\top x + f_i \end{pmatrix} = \mu, \quad i = 1, ..., s.$$

2. (x, σ, λ, v) have to satisfy

$$\|D_i x + d_i\|_2 < e_i^\top x + f_i \quad \text{and} \quad \|v_i\|_2 < \lambda_i \;\; i = 1, ..., s.$$

This means that x and (σ, λ, v) have to lie in the (relative) interior of the primal and dual feasible set, respectively. This motivates the name interior-point method.

Of course there are several variants of primal-dual interior-point methods. A general outline of these methods is as follows (see also Figure 2.3).

1. They start with a point that lies near (or on) the central path.

2. Then they apply Newton's method to the equalities in the system (2.24) to get a direction $(\Delta x, \Delta\sigma, \Delta\lambda, \Delta v)$ that reduces the duality gap. The step length is chosen such that the new iterate is still feasible and satisfies the strict inequalities

$$\|D_i x + d_i\|_2 < e_i^\top x + f_i \quad \text{and} \quad \|v_i\|_2 < \lambda_i \;\; i = 1, ..., s.$$

Hence, each iterate of the methods lies in the (relative) interior of the feasible sets.

3. After each iteration the duality gap μ is reduced by a constant factor. This means the methods follow the central path for a decreasing value of μ where $\mu = 0$ corresponds to optimality.

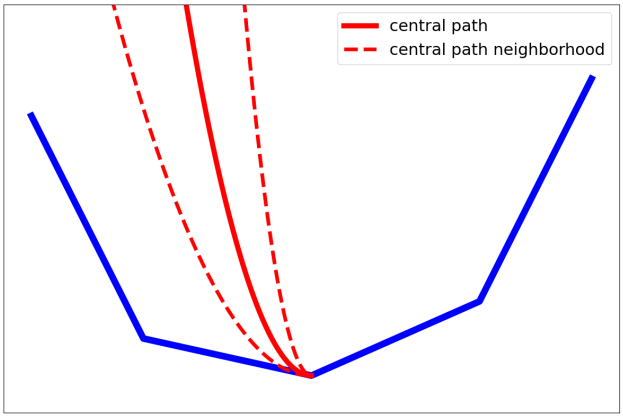

Figure 2.3: Illustration of a interior-point method. Iterates lie on the central path or in some neighborhood. From Nocedal and Wright 2006, p.400

With an appropriate choice of the initial point, step length and reduction schedule for μ one can show convergence in a polynomial number of iterations. Moreover, these algorithms do not only provide theoretical bounds, but they are also useful in practice.

2.2.6 Example: Random objective function

Literature: Prékopa 1995, section 8.5

As mentioned in Section 2.1 a direct use of single chance constraints is usually not a good idea from the modeling point of view. Accordingly, there are not too many examples where an immediate application of single chance constraints seems to be reasonable. However, one such example are problems where all the problem data is assumed to be deterministic except for a random objective function $c(\xi)^\top x$. Therefore, we consider the following problem:

$$\begin{aligned} \text{minimize} \quad & c(\xi)^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

Similar as for chance constraints we first have to define what is meant by a random objective function. Again there are several possibilities. The approach we choose here is related to chance constraints. That is, we introduce an auxiliary variable t and the put objective function into a new constraint, i.e.

$$\begin{aligned} \text{minimize} \quad & t \quad \text{where } x \in \mathbb{R}^n, t \in \mathbb{R} \\ \text{subject to} \quad & \mathbb{P}(c(\xi)^\top x \leq t) \geq \alpha \\ & Ax = b \\ & x \geq 0. \end{aligned}$$

Hence, in this problem we want to find a solution vector x^* and a minimum number t^* such that the objective value $c(\xi)^\top x^*$ is smaller equal than this number t^* with a (high) probability α. This approach naturally leads to a problem with exactly one single chance constraint. If $c(\xi)$ has a multivariate normal distribution, the theory of the previous sections can be applied. If not, approximation strategies for single chance constraints can be used, see also Remark 2.5.

2.3 Joint chance constraints - Random right hand side

In this section we consider the joint chance constraint problem (JCCP) in the special case when only the right hand side is random, i.e.

$$T(\xi) \equiv T \text{ deterministic} \quad \text{and} \quad h(\xi) = \xi \in \mathbb{R}^s.$$

Additionally, we assume that ξ has a nondegenerate multivariate normal distribution, i.e.

$$\xi \sim \mathcal{N}(\mu, \Sigma)$$

with Σ being a positive definite matrix. Then ξ has the density function

$$f(y) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(\Sigma)}} e^{-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)}.$$

We consider the following chance constraint optimization problem in this section:

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \\ & \mathbb{P}(Tx \geq \xi) \geq \alpha. \end{aligned} \tag{2.25}$$

We present a cutting plane method for solving (2.25). We introduce the method for a more general type of problems. However, it turns out that in the general case some steps of the algorithm cannot be ensured to always work (namely the availability of a strictly feasible point and the guarantee that the gradients do not vanish at boundary points). In Section 2.3.2 we show that these obstacles do not occur when solving (2.25) and explain why this method is particularly well suited for (2.25).

2.3.1 A cutting plane method

In order to be able to formulate the general problem we introduce the notion of a quasiconcave function.

Quasiconcave functions

Literature: Bazaraa, Sherali, and Shetty 2006, p.134-138

Definition 2.17 (Quasiconcave function)
Let $C \subset \mathbb{R}^n$ be a convex set. A function $g : C \to \mathbb{R}$ is called quasiconcave if for all $x, y \in C$

$$g(\lambda x + (1-\lambda)y) \geq \min(g(x), g(y)) \quad \forall \lambda \in [0,1] \tag{2.26}$$

holds.

Evidently, every concave function is also quasiconcave. Moreover, one can show easily that quasiconcavity is equivalent to the property of a function that all its upper level sets are convex, i.e.

$$g \text{ quasiconcave} \iff \{x \mid g(x) \geq \gamma\} \text{ is convex} \quad \forall \gamma \in \mathbb{R}.$$

Therefore, quasiconcavity is a reasonable generalization of concavity. However, the example $f(x) = x^3$ shows that there are quasiconcave functions that are not concave.

The important property of quasiconcave functions for the method which is presented in this section is the following:

Lemma 2.18 (from Bazaraa, Sherali, and Shetty 2006, theorem 3.5.4)
Let $C \subset \mathbb{R}^n$ be a convex set and $g : C \to \mathbb{R}$ a differentiable quasiconcave function. Then it holds for all $x, y \in C$:

$$g(x) \geq g(y) \implies \nabla g(y)(x - y) \geq 0. \tag{2.27}$$

Proof. Let $x, y \in C$ such that $g(x) \geq g(y)$. Applying Taylor expansion results in

$$g(\lambda x + (1-\lambda)y) - g(y) = \lambda \nabla g(y)(x - y) + o(\lambda) \quad \forall \lambda \in [0, 1].$$

By the quasiconcavity of g it holds $g(\lambda x + (1-\lambda)y) \geq g(y)$ which implies

$$\lambda \nabla g(y)(x - y) + o(\lambda) \geq 0 \quad \forall \lambda \in [0, 1].$$

Dividing by λ and letting $\lambda \to 0$ shows that $\nabla g(y)(x - y) \geq 0$. □

Motivation of the method

With this property in mind we consider the following problem:

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \\ & g_i(x) \geq 0, \ i = 1, ..., l \end{aligned} \tag{2.28}$$

where $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., l$, are quasiconcave differentiable functions. We assume that the set

$$P_0 := \{x \mid Ax = b, x \geq 0\} \tag{2.29}$$

is bounded. This can be enforced by box constraints $0 \leq x_i \leq u_i, i = 1, ..., n$. This implies also the boundedness of the feasible convex set

$$C := \{x \mid Ax = b, x \geq 0, g_i(x) \geq 0 \ \forall i = 1, ...l\}.$$

Since our actual goal is to solve (2.25), it is helpful to think about the special case of (2.28) with

$$l = 1 \quad \text{and} \quad g_1(x) = \mathbb{P}(Tx \geq \xi).$$

We show in Section 2.3.2 that $\mathbb{P}(Tx \geq \xi)$ is quasiconcave in x (in fact it satisfies a stronger property which is important for the method to work).

(2.28) is solved by a cutting plane method. We will outline now the idea of cutting plane methods (see Figure 2.4). (2.28) is about minimizing a linear function over a convex set C. A cutting plane method sequentially solves linear programming relaxations of this problem, i.e. builds outer polyhedral approximations P of the feasible set C. If the optimal solution x^* of the relaxation (e.g. a vertex found by the simplex method) is also feasible for the original problem, then it is of course also optimal for

the original problem. If not, a linear inequality $h^\top x \leq z$ (a.k.a. a cutting plane) is added that cuts off the optimal relaxed solution x^* but no feasible point of the original problem, i.e.

$$h^\top x^* > z$$

but

$$h^\top x \leq z \quad \forall x \in C.$$

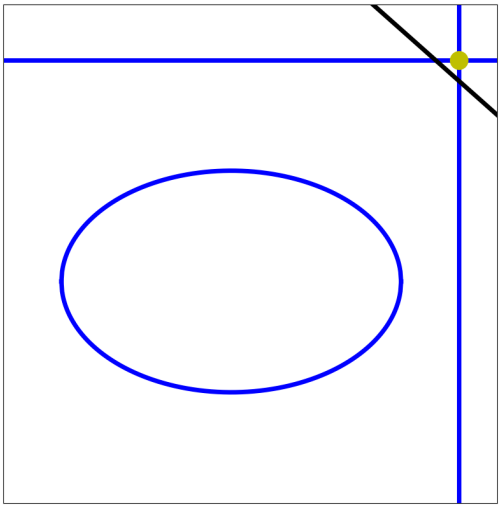

Figure 2.4: The feasible set is a convex ellipse. The polyhedral approximation consists of box constraints for two the variables. The optimal vertex of this approximation (yellow) is cut off by a hyperplane (black).

In Figure 2.4 two possible drawbacks of this general procedure become apparent:

1. On the one hand a cutting plane can be arbitrary bad in the sense that it cuts off the optimal solution but almost nothing else of the polyhedron. This can make the method quite slow.

2. On the other hand in practice this method has to be stopped after a finite number of iterations. It might happen that until then no feasible point was found. Of course the relaxed solutions provide lower bounds for the optimal value but no upper bounds are available. Hence, one cannot estimate how good (or bad) this lower bound is. Moreover, it is not clear which point should be returned as an (approximate) solution if the method only detected infeasible points so far.

The cutting plane method presented here will overcome these two problems (see Figure 2.5 and Figure 2.6):

1. To avoid problem 1 supporting hyperplanes are generated. These are cutting planes that ”touch” the feasible set. Therefore, the presented method is also called a supporting hyperplane method.

2. Problem 2 is addressed by a sequence of boundary points that are constructed along the method by the use of an initially given interior point. As they are feasible for the original problem they provide upper bounds on the optimal value.

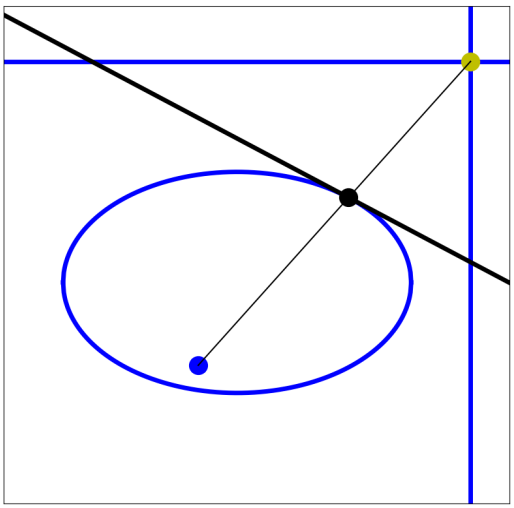

Figure 2.5: The cutting plane (black) supports the feasible set at a bounday point point (black). The boundary point lies on the line between a strictly feasible point (blue) and the optimal relaxed solution (yellow).

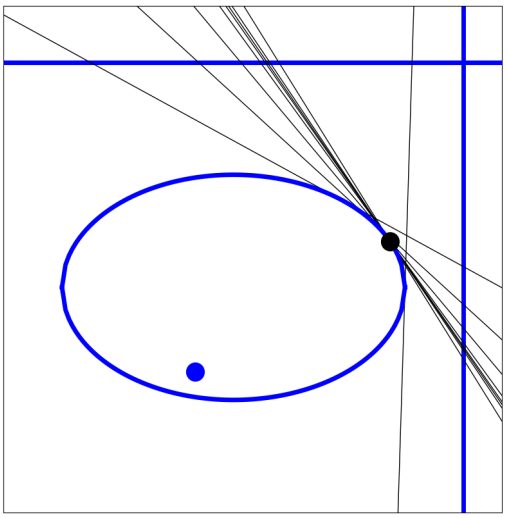

Figure 2.6: Visualization of the termination of a computer program for the supporting hyperplane method. The black point is the boundary point with the smallest objective value and is returned as an optimal solution.

Supporting hyperplane method of Veinott

Literature: Veinott 1967

Now we derive the method formally. It is summarized in Algorithm 1.

First of all, the method needs a strictly feasible point a as an input, i.e. a is feasible for (2.28) and

$$g_i(a) > 0 \quad \forall\, i = 1, ..., l. \tag{2.30}$$

The initial poylyhedral relaxation of the feasible set C is the linearly constrained set P_0. Therefore in iteration $k = 0$ the problem

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{2.31}$$

is solved. If the optimal solution x_k^* is already feasible for the original problem (i.e. $x_k^* \in C$), then x_k^* is also optimal and the algorithm stops.

If not, we construct a linear inequality that cuts off x_k^* but no point of the feasible set C. In order to do this, we determine a point on the line betweeen the strictly feasible point a and the relaxed solution x_k^*, i.e. we find the largest number λ_k such that

$$b_k := a + \lambda_k(x_k^* - a) \in C.$$

We know for sure that $0 < \lambda_k < 1$ as a is strictly feasible for C, x_k^* is infeasible for C and C is a convex set. λ_k can be computed with a bisection method. Here the interval $(0, 1)$ is successively halved and the constraints are evaluated at the center of the interval. If the center is feasible, then the right half of the interval is chosen in the next step, else the left half. We mention that the bisection method can be sped up if there are bounds available for the functions g_j, $j = 1, ..., l$. This applies to probability functions, see also Remark 2.30.

The point b_k lies on the relative boundary of C since otherwise λ_k could be increased. Moreover, there is an index $j \in \{1, ..., l\}$ such that

$$g_j(b_k) = 0.$$

For every feasible point $x \in C$ we have

$$g_j(x) \geq 0 = g_j(b_k).$$

Then Lemma 2.18 implies that

$$\nabla g_j(b_k)(x - b_k) \geq 0 \quad \forall x \in C. \tag{2.32}$$

This inequality is of course only useful if

$$\nabla g_j(b_k) \neq 0$$

holds. In Lemma 2.20 we show that then this inequality indeed cuts off x_k^*. However, $\nabla g_j(b_k) \neq 0$ cannot always be ensured for quasiconcave functions and the algorithm might get stuck at this point. Therefore one has to guarantee somehow that

$$g_j(b_k) = 0 \implies \nabla g_j(b_k) \neq 0.$$

We show in Section 2.3.2 how this can be ensured for the joint chance constraint $\mathbb{P}(Tx \geq \xi)$.

If $\nabla g_j(b_k) \neq 0$ holds, then

$$\{x \mid \nabla g_j(b_k)(x - b_k) = 0\}$$

defines a supporting hyperplane to C as $\nabla g_j(b_k)(b_k - b_k) = 0$ and $C \subset \{x \mid \nabla g_j(b_k)(x - b_k) \geq 0\}$. In this case we refine the polyhedral approximation P_k, i.e.

$$P_{k+1} := P_k \cap \{x \mid \nabla g_j(b_k)(x - b_k) \geq 0\}$$

and solve (2.31) again with P_{k+1} as the feasible set. In Algorithm 1 we also assume that $\nabla g_j(b_k) \neq 0$ holds throughout the algorithm.

Algorithm 1 Supporting hyperplane method (Veinott)

1: **Input**: A problem of type (2.28) and a feasible point for (2.28) satisfying

$$g_i(a) > 0 \quad \forall\, i = 1, ..., l$$

2: **Output**: Optimal solution x^* for (2.28)
3: **Initialization**: Set $k = 0$ (P_0 is defined in (2.29))
4: Solve the LP relaxation

$$\begin{aligned} &\text{minimize} \quad c^\top x \quad \text{where } x \in \mathbb{R}^n \\ &\text{subject to} \quad x \in P_k \end{aligned} \tag{2.33}$$

5: Let x_k^* be the optimal solution.
6: **if** $x_k^* \in C$ **then**
7: stop and return x_k^* which is optimal for (2.28)
8: **else if** some stopping criterion is met **then**
9: stop and return b_v for

$$v = \underset{i=1,\ldots,k}{\arg\min}\, c^\top b_i$$

10: **else**
11: Find the largest number $\lambda_k \in [0, 1]$ such that $b_k := a + \lambda_k(x_k^* - a) \in C$.
12: Then b_k lies on the boundary of C and

$$\exists\, j \in \{1, ...l\} \text{ such that } g_j(b_k) = 0$$

13: Add the linear inequality

$$\nabla g_j(b_k)(x - b_k) \geq 0 \tag{2.34}$$

14: to the relaxed feasible set, i.e.

$$P_{k+1} := P_k \cap \{x \mid \nabla g_j(b_k)(x - b_k) \geq 0\}$$

15: k := k+1, Goto 3
16: **end if**

Remark 2.19

1. *The feasible set in (2.33) differs from the feasible set of the previous iteration only by the added inequality (2.34). Hence, the dual simplex method is particularly well suited to solve these problems.*

2. *The relaxed solutions x_k^* provide increasing lower bounds on the optimal value of (2.28). The boundary points b_k provide (not necessarily decreasing) upper bounds on the optimal value of (2.28). Hence, after iteration k the estimation*
$$\min_{i=1,\dots,k} c^\top b_i - c^\top x_k^*$$
is available as an upper bound for the optimality gap and can be used as a practical stopping criterion.

Nevertheless, two important questions still remain open:

1. How to find the point a in (2.30)? We will answer this question in Section 2.3.2.

2. What are the convergence properties of this method? This will be answered now.

Convergence properties of the supporting hyperplane method

Literature: Veinott 1967

To prove the convergence let
$$N := \{x \mid g_i(x) \geq 0 \; \forall \; i = 1, ..., l\}$$
be the feasible set of the nonlinear constraints. The convergence proof focuses on the set N rather than the set $C \subset N$. The reason for this is that all the linear constraints of C are satisfied in each iteration of the algorithm. Moreover, the strictly feasible point a lies in the interior of N rather than the relative interior of C which is important for the proof to work.

We assume that
$$\nabla g_j(b_k) \neq 0$$
always holds and hence in each iteration a supporting hyperplane can be found as outlined above. In Section 2.3.2 we show that this always holds for problem (2.25). Following the notation of Veinott 1967 we write the supporting hyperplane inequality in iteration k as
$$h_k^\top x \leq z_k.$$
It holds
$$N \subset H_k := \{x \mid h_k^\top x \leq z_k\},$$
and
$$h_k^\top b_k = z_k$$
holds for a boundary point
$$b_k = a + \lambda_k(x_k^* - a).$$

We note that in the notation of (2.32) it holds:

$$h_k = -\nabla g_j(b_k) \text{ and } z_k = -\nabla g_j(b_k) b_k.$$

Moreover, we observe that (2.32) is also a supporting hyperplane to N.

We prove the convergence of the supporting hyperplane method in a slightly more general setting that does not depend on the explict construction of the supporting hyperplane but only on the existence of a linear inequality that defines a supporting hyperplane to N.

The presentation starts with a lemma showing that a supporting hyperplane indeed cuts off the relaxed solution x_k^* from the relaxed feasible set P_k.

Lemma 2.20 (from Veinott 1967, lemma 3)
Let x_k^* be any iterate of Algorithm 1. If $x_k^* \notin N$, then $x_k^* \notin H_k$.

Proof. $\{x \mid h_k^\top x = z_k\}$ intersects N only at boundary points (otherwise there is $x \in N$ such that $h_k^\top x > z_k$). Hence, $h_k^\top a < z_k$. Since $a \in \text{int(N)}$ and $x_k^* \notin N$ it holds: $0 < \lambda_k < 1$.
Moreover

$$\begin{aligned} z_k = h_k^\top b_k &= h_k^\top((1-\lambda_k)a + \lambda_k x_k^*) \\ &= (1-\lambda_k) \underbrace{h_k^\top a}_{<z_k} + \lambda_k h_k^\top x_k^*. \end{aligned}$$

This implies $h_k^\top x_k^* > z_k$ and hence x_k^* is cut off. □

The next statement will also be needed in the convergence proof.

Lemma 2.21 (from Veinott 1967, theorem 1)
Let x^* be an accumulation point of the iterates $\{x_k^*\}$ of Algorithm 1 such that $x^* \in C$. Then x^* is optimal for (2.28).

Proof. Evidently, $P_0 \supset P_1 \supset \ldots \supset C$ and hence with f^* being the optimal value of (2.28)

$$c^\top x_1^* \leq c^\top x_2^* \leq \ldots \leq f^*.$$

By assumption $x^* \in C$ (i.e. is feasible for (2.28)). As linear functions are continous it also holds $c^\top x^* \leq f^*$. Hence, x^* is optimal. □

Now we can prove the main convergence theorem.

Theorem 2.22 (Convergence of the supporting hyperplane method, from Veinott 1967, theorem 2) *If $x_k^* \notin C$ for all k, then the accumulation points of $\{x_k^*\}$ and $\{b_k\}$ of Algorithm 1 coincide and are optimal for problem (2.28).*

Proof. Since C is closed and $b_k \in C$ for all k, all the accumulation points of $\{b_k\}$ lie in C. Therefore, it suffices to show that the accumulation points of $\{x_k^*\}$ and $\{b_k\}$ coincide as then all the accumulation points of $\{x_k^*\}$ lie in C and the statement follows by Lemma 2.21.

W.l.o.g. we assume that $\|h_k\| = 1 \; \forall k$, i.e. h_k is bounded (otherwise we divide each h_k by $\|h_k\|$).

Let x^* be an accumulation point of $\{x_k^*\}$. It exists as $x_k^* \in P_0$ for all k and we assumed P_0 to be bounded. Moreover the sequences $\{b_k\} \subset C \subset P_0$, $\{h_k\}$ and $\{\lambda_k\} \subset (0,1)$ are bounded. Applying the theorem of Bolzano-Weierstrass several times implies that there exists a subsequence such that

$$\lim_{i \to \infty} (x_i, b_i, h_i, \lambda_i) = (x^*, b^*, h^*, \lambda^*)$$

for some vectors x^*, b^*, h^*, λ^*. Now we show $x^* = b^*$.

It holds

$$h_i x \leq z_i = h_i b_i \quad \forall i \; \forall x \in N$$

as $N \subset H_i \; \forall i$. Letting $i \to \infty$ leads to

$$h^* x \leq h^* b^* \quad \forall x \in N. \tag{2.35}$$

By construction of the algorithm $x_{i+1} \in H_i$ and according to Lemma 2.20 $x_i \notin H_i$.

This implies

$$h_i x_{i+1} \leq h_i b_i < h_i x_i \quad \forall i$$

and in the limit $i \to \infty$

$$h^* x^* = h^* b^*.$$

Using $b^* = (1 - \lambda^*)a + \lambda^* x^*$ leads to

$$h^* x^* = h^* b^* = (1 - \lambda^*) h^* a + \lambda^* h^* x^*.$$

This implies

$$(1 - \lambda^*) h^* b^* = (1 - \lambda^*) h^* x^* = (1 - \lambda^*) h^* a. \tag{2.36}$$

As $a \in \text{int(N)}$, $h^* a < h^* b^*$ since otherwise there is $x \in N$ such that $h^* x > h^* b^*$ contradicting (2.35). Then $\lambda^* = 1$ because of (2.36) and hence $b^* = x^*$. This shows that each accumulation point of $\{x_k^*\}$ is also one of $\{b_k\}$. For the inverse implication exchange the roles of x^* and b^*.

□

2.3.2 Application of the supporting hyperplane method to joint chance constraint problems

In this section we apply Algorithm 1 to problem (2.25). This section is divided in three parts:

1. The (generalized) concavity properties of the constraint

$$g(x) := \mathbb{P}(Tx \geq \xi) \geq \alpha \quad \text{with} \quad \xi \sim \mathcal{N}(\mu, \Sigma).$$

   Related with this is also the question how $\nabla g(b_k) \neq 0$ can be ensured in (2.32).

2. The problem (2.30) of finding a point $a \in C$ satisfying

$$g(a) = \mathbb{P}(Ta \geq \xi) > \alpha.$$

3. The evaluation of $\mathbb{P}(Tx \geq \xi)$ and $\nabla\mathbb{P}(Tx \geq \xi)$.

Generalized concavity properties of probability functions

In Section 2.3.1 we introduced quasiconcavity as a generalization of concavity. In the following we will introduce two further generalizations of concavity along with their relationships and meanings for solving problem (2.25) (see Figure 2.7):

$$\text{logconcave} \implies \text{pseudoconcave} \implies \text{quasiconcave}$$

Moreover we make use of theorem about the relation of logconcave probability density functions and their corresponding probability measures.

The presentation starts with the definition of a logconcave function.

Definition 2.23 (Logconcave function)
Let $C \subset \mathbb{R}^n$ be a convex set. A function $f : C \to \mathbb{R}_+$ is called logconcave if for all $x, y \in C$ and for all $\lambda \in [0,1]$

$$\log(f(\lambda x + (1-\lambda)y)) \geq \lambda \cdot \log(f(x)) + (1-\lambda) \cdot \log(f(y))$$

holds.

The definition basically says that f is logconcave if and only if $\log(f)$ is a concave function. Moreover one can show that a positive concave function is also logconcave.

The notion of a logconcave probability measure is defined similarly.

Definition 2.24 (Logconcave measure)
A probability measure $\mathbb{P}$ is called logconcave if for all pair of convex sets $A, B \subset \mathbb{R}^n$ and for all $\lambda \in [0,1]$

$$\log(\mathbb{P}(\lambda A + (1-\lambda)B)) \geq \lambda \log(\mathbb{P}(A)) + (1-\lambda)\log(\mathbb{P}(B))$$

holds.

Now we can formulate without proof the aforementioned theorem.

Theorem 2.25 (from Prékopa 1995, theorem 4.2.1)
Let f be a logconcave probability density function and $\mathbb{P}$ be the probability measure generated by f. Then $\mathbb{P}$ is a logconcave measure.

To see how this theorem applies to the setting of (2.25) we consider the probability density function of the right hand side vector ξ which was assumed to be nondegenerate multivariate normal distributed:

$$f(y) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(\Sigma)}} \cdot e^{-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)}.$$

Taking the logarithm on both sides and neglecting an additive constant leads to

$$\log(f(y)) = -\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu).$$

As Σ was assumed to be positive definite, the same holds for Σ^{-1}. Therefore $\log(f(y))$ is concave and hence $f(y)$ a logconcave density function. Theorem 2.25 implies that the corresponding measure $\mathbb{P}$ is also logconcave.

Setting

$$g(x) := \mathbb{P}(Tx \geq \xi),$$

and

$$A := \{\xi \mid Tx \geq \xi\}, B := \{\xi \mid Ty \geq \xi\}$$

for fixed vectors x and y, we see that g is a logconcave function since

$$\begin{aligned}
\log(g(\lambda x + (1-\lambda)y)) &= \log(\mathbb{P}(T(\lambda x + (1-\lambda)y) \geq \xi)) \\
&= \log(\mathbb{P}(\lambda A + (1-\lambda)B)) \\
&\geq \lambda \log(\mathbb{P}(A)) + (1-\lambda)\log(\mathbb{P}(B)) \\
&= \lambda \log(\mathbb{P}(Tx \geq \xi)) + (1-\lambda)\log(\mathbb{P}(Ty \geq \xi)) \\
&= \lambda \log(g(x)) + (1-\lambda)\log(g(y)).
\end{aligned}$$

Remark 2.26
The logconcavity of $\mathbb{P}(Tx \geq \xi)$ does not only hold if ξ follows a nondegenerate multivariate normal distribution. The important property of the normal distribution is its logconcave probability density function which is by Theorem 2.25 sufficient (though not necessary) for the logconcavity of the corresponding probability measure. For further examples of logconcave probability distributions see Prékopa 1995, section 4.4.

Next, we show that every logconcave function is pseudoconcave.

Definition 2.27 (Pseudoconcave function)
Let $C \subset \mathbb{R}^n$ be a convex set. A differentiable function $g : C \to \mathbb{R}$ is called pseudoconcave if for all $x, y \in C$

$$\nabla g(x)(y - x) \leq 0 \implies g(y) \leq g(x)$$

holds.

Again, every concave function is also pseudoconcave by the gradient characterization of concavity (see Herzog 2023, theorem 13.17 ii)). Moreover we note that by the definition it is obvious that for a pseudoconcave function $\nabla g(x) = 0$ is sufficient for x to be a global maximimizer.

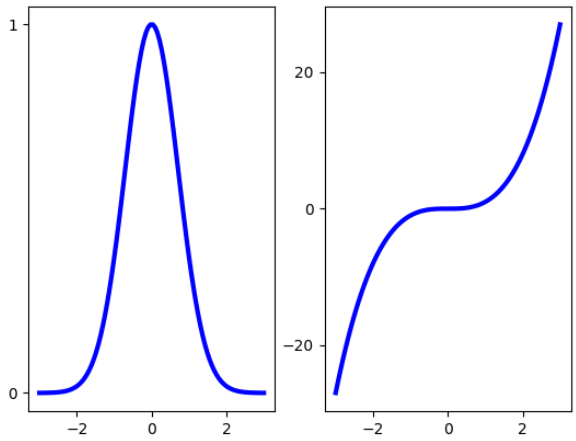

Figure 2.7: On the left the function e^{-x^2} is plotted. It is both logconcave and pseudoconcave, but not concave. On the right the function x^3 is plotted. It is quasiconcave but not pseudoconcave as the gradient vanishes at $x = 0$ but there is no maximum.

Lemma 2.28 (from Kall and Mayer 2010, proposition 2.35)
Let g be a positive, differentiable and logconcave function over a convex set $C \subset \mathbb{R}^n$. Then g is also pseudoconcave.

Proof. Let $x, y \in C, \lambda \in [0, 1]$ and assume $\nabla g(x)(y - x) \leq 0$. This implies

$$\nabla(\log(g(x)))(y - x) = \frac{1}{g(x)} \nabla g(x)(y - x) \leq 0. \tag{2.37}$$

By assumption $\log(g(x))$ is concave and hence also pseudoconcave. Considering (2.37) this implies

$$\log(g(y)) \leq \log(g(x))$$

and hence $g(y) \leq g(x)$ as the logarithm is monotone. □

We conclude that $g(x) = \mathbb{P}(Tx \geq \xi)$ is a pseudoconcave function. This has an important consequence for Algorithm 1: In (2.32) there was the restriction that the gradient must not vanish at boundary points. Given that there is a point a satisfying

$$\mathbb{P}(Ta \geq \xi) > \alpha \tag{2.38}$$

this always holds: Assume by contradiction that $\nabla\mathbb{P}(Tb \geq \xi) = 0$ for some boundary point b (i.e. $\mathbb{P}(Tb \geq \xi) = \alpha$). This would mean that b is a global maximimizer contradicting the existence of a point a in (2.38).

Finally, we show that pseudoconcavity implies quasiconcavity.

Lemma 2.29 (from Bazaraa, Sherali, and Shetty 2006, theorem 3.5.11 and lemma 3.5.7)
Let $g : C \to \mathbb{R}$ be a pseudoconcave function over a convex set $C \subset \mathbb{R}^n$. Then g is quasiconcave.

Proof. Let $x, y \in C$. In this proof let $x(\lambda) := \lambda x + (1-\lambda)y$ for $\lambda \in (0,1)$ (see Figure 2.8). The proof consists of two cases.

Case 1:
For $g(x) \neq g(y)$ we prove a statement slightly stronger than quasiconcavity:

$$g(x) \neq g(y) \implies g(x(\lambda)) > \min(g(x), g(y)) \quad \forall \lambda \in (0,1).$$

We assume by contradiction that $g(x) \neq g(y)$ and there is $x(\lambda)$ such that $g(x(\lambda)) \leq \min(g(x), g(y))$. W.l.o.g. let $g(x) > g(y)$ and hence $g(x(\lambda)) \leq g(y) < g(x)$. The pseudoconcavity of g then implies

$$\nabla g(x(\lambda))(x - x(\lambda)) > 0.$$

It is easy to see that $x - x(\lambda) = -(1-\lambda)\frac{(y-x(\lambda))}{\lambda}$ which implies

$$\nabla g(x(\lambda))(y - x(\lambda)) < 0.$$

Again by the pseudoconcavity of g it follows $g(y) \leq g(x(\lambda))$ and hence $g(y) = g(x(\lambda))$.

$\nabla g(x(\lambda))(y - x(\lambda)) < 0$ implies the existence of a point $x(\mu)$ on the line between $x(\lambda)$ and y such that

$$g(x(\mu)) < g(x(\lambda)) = g(y).$$

The pseudoconcavity of g implies

$$\nabla g(x(\mu))(y - x(\mu)) > 0$$

and

$$\nabla g(x(\mu))(x(\lambda) - x(\mu)) > 0.$$

It also holds $y - x(\mu) = \mu\frac{x(\mu)-x(\lambda)}{1-\mu}$ which contradicts the above two inequalities.

Case 2:
Suppose $g(x) = g(y)$. Assume by contradiction that $g(x(\lambda)) < g(x)$. Then there exists a point $x(\sigma)$ on the line between x and $x(\lambda)$ such that

$$g(x(\lambda)) < g(x(\sigma)) < g(x) = g(y). \tag{2.39}$$

We note that $x(\lambda)$ can be written as a convex combination of $x(\sigma)$ and y. Observing $g(y) > g(x(\sigma))$ and applying Case 1 implies $g(x(\lambda)) > g(x(\sigma))$. But this contradicts (2.39).

□

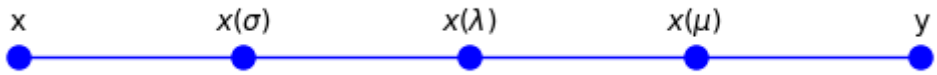

Figure 2.8: Visualization of the points occurring in the proof of Lemma 2.29

Hence, $g(x) = \mathbb{P}(Tx \geq \xi)$ is also quasiconcave and we can apply Algorithm 1. As shown above the problem that the gradient vanishes in (2.34) does not occur due to the pseudoconcavity of $\mathbb{P}(Tx \geq \xi)$ and the existence of a point which is strictly feasible for the chance constraint. We will now consider the problem of finding such a point.

Finding the initial interior point

Literature: Kall and Mayer 2010, section 4.3.2

In (2.30) in Algorithm 1 a point a that strictly satisfies all nonlinear constraints is required. In the case of (2.25) this simplifies to the existence of a point a satisfying

$$\mathbb{P}(Ta \geq \xi) > \alpha$$

as there is only exactly one nonlinear constraint. For this special case the following phase-I-problem can be solved:

$$\begin{aligned} \text{maximize} \quad & \mathbb{P}(Tx \geq \xi) \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

We can write it equivalently as

$$\begin{aligned} \text{maximize} \quad & t \quad \text{where } x \in \mathbb{R}^n, t \in \mathbb{R} \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \\ & \log(\mathbb{P}(Tx \geq \xi)) - t \geq 0. \end{aligned} \tag{2.40}$$

Here $\log(\mathbb{P}(Tx \geq \xi))$ is concave and hence the entire constraint is concave. (2.40) can be solved with Algorithm 1 as well:

1. Find a feasible point x for the linear constraints, i.e. $Ax = b, x \geq 0$. This can be done by solving a linear program.

2. Choose t such that
$$\log(\mathbb{P}(Tx \geq \xi)) - t > 0.$$
Then (x, t) can be used as a strictly feasible point to start Algorithm 1 for (2.40). We note that the optimization needn't be executed until convergence. It suffices to stop when a feasible point $(\overline{x}, \overline{t})$ is found that satisfies $\overline{t} > \log(\alpha)$ which implies
$$\mathbb{P}(T\overline{x} \geq \xi) > \alpha.$$

Evaluating probability expressions

First, we consider the evaluation of the expression $\mathbb{P}(Tx \geq \xi)$ for nondegenerate $\xi \sim \mathcal{N}(\mu, \Sigma)$ and fixed x. Using the normal density function it can be computed as

$$\mathbb{P}(Tx \geq \xi) = \frac{1}{\sqrt{(2\pi)^n \cdot \det(\Sigma)}} \int_{\{y \leq Tx\}} e^{-\frac{1}{2}(y-\mu)^\top \Sigma^{-1}(y-\mu)}\, dx. \tag{2.41}$$

However, numerical methods for computing this expression become very inefficient in high dimensions. A remedy is to use Monte-Carlo methods to approximate this expression. A detailed discussion of these methods is far beyond the scope of this thesis. Hence, we give only a rough outline of the method. We consider the integral

$$\int_A f(x)\, dx \tag{2.42}$$

and assume that $A \subset \mathbb{R}^n$ is bounded (in (2.41) this can be achieved by an integral transformation).

A Monte-Carlo method then works as follows:

1. Pick random elements $x_1, ..., x_N \in A$

2. For $\lambda(A)$ being the Lebesgue measure of A compute the average
$$\lambda(A)\frac{1}{N}\sum_{i=1}^{N} f(x_i)$$
as an approximation of (2.42).

Of course, there are several improvements available. Further details can be found in Kall and Mayer 2010, section 4.3.5.

Remark 2.30 (Bounds on probability functions)
Evaluating probability functions in high dimensions with Monte-Carlo simulation is computationally expensive. However, in the bisection method of Algorithm 1 many evaluations of the probability function

are necessary to find the boundary point b_k with a reasonable accuracy. Therefore, one strives to reduce the number of required function evaluations without losing accuracy.

This is accomplished by the use of bounds for probability functions. We set $F(x) := \mathbb{P}(Tx \geq \xi)$. Let F_L and F_U be lower and upper bounds for F, respectively, i.e.

$$F_L(x) \leq F(x) \leq F_U(x) \quad \forall x \in \mathbb{R}^n.$$

We consider the first iteration ($\lambda = \frac{1}{2}$) of the bisection method to illustrate the use of these bounds. We know that

$$\mathbb{P}(Ta \geq \xi) > \alpha,$$
$$\mathbb{P}(Tx_k^* \geq \xi) < \alpha$$

and want to find a point $b_k = a + \lambda(x_k^* - a)$ such that

$$\mathbb{P}(Tb_k \geq \xi) = \alpha.$$

If

$$F_U(a + \frac{1}{2}(x_k^* - a)) < \alpha$$

holds, we can choose $[\frac{1}{2}, 1]$ as the next interval. If

$$F_L(a + \frac{1}{2}(x_k^* - a)) > \alpha$$

holds, we can choose $[0, \frac{1}{2}]$ as the next interval.

If none of these two inequalities hold, we have to use Monte-Carlo simulation. But in the other two cases it is possible to halve the interval without using computationally expensive Monte-Carlo simulation. Fortunately, there are good lower and upper bounds available that are relatively easy to compute. For further details see Kall and Mayer 2010, section 4.3.4, Prékopa 1995, sections 6.1-6.3, and Szántai 1988.

Finally, we briefly consider the computation of derivatives of probability expressions. One can show that

$$\nabla \mathbb{P}(Tx \geq \xi)$$

is again multivariate normal distributed and consequently the same methods as for evaluating $\mathbb{P}(Tx \geq \xi)$ can be used. For details see Prékopa 1995, section 6.6.4

2.3.3 Example: Transport problem with random demands

We consider a transport network which is represented as a graph. There are nodes that supply a commodity ("suppliers") and there are nodes that demand the commodity ("customers"), see Figure 2.9. The network structure is represented by a matrix $A = (A_{\text{supply}}, A_{\text{demand}})$, the supplies and demands by a vector $b = (b_{\text{supply}}, b_{\text{demand}})$. The costs of transporting a unit of the commodity over an edge are stored in the cost vector c. The solution vector x indicates which amount of the commodity should be shipped over each edge. Furthermore, there are box constraints $0 \leq x \leq u$ that restrict the amount of

the commodity that can be transported over each edge (compare Herzog 2023, §11).
In a deterministic framework this problem can be formulated as

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & A_{\text{supply}} x \leq b_{\text{supply}} \\ & A_{\text{demand}} x \geq b_{\text{demand}} \\ & 0 \leq x \leq u. \end{aligned} \tag{2.44}$$

However, in reality the problem data is often not deterministic. Here we consider the case when the customer demands are random, i.e.

$$b_{\text{demand}} = b(\xi) \quad \text{with} \quad \xi \sim \mathcal{N}(\mu, \Sigma).$$

In practice, we usually cannot wait for the realization of the randomness (the actual demands) as there might be a time delay between the start of transportation and the arrival at the customers. To deal with this, we formulate (2.44) as a chance constraint problem of type (2.25):

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & A_{\text{supply}} x \leq b_{\text{supply}} \\ & 0 \leq x \leq u \\ & \mathbb{P}(A_{\text{demand}} x \geq b(\xi)) \geq \alpha \end{aligned}$$

This problem ensures that all the customer demands as a whole are satisfied with a probability of at least α. It can be solved by the supporting hyperplane method of the previous section.

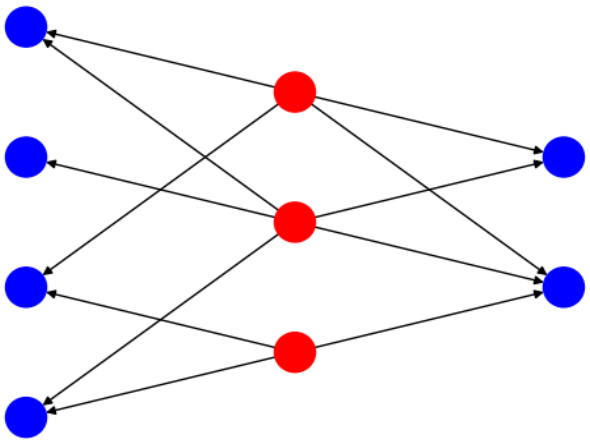

Figure 2.9: Transport network. The red nodes correspond to suppliers, the blue nodes to customers.

2.4 Joint chance constraints - The general case

In this section we consider the joint chance constraint

$$\mathbb{P}(T(\xi)x \geq h(\xi)) \geq \alpha$$

in its full generality, i.e. T and h are both allowed to be random.

These constraints are very difficult to deal with. To the best of my knowledge there is only one special case where the convexity of the feasible set can be ensured. However, this special case requires very strong assumptions on the distribution and is consequently not of practical use to design algorithms. Details can be found in Prékopa 1995, theorem 10.4.3 and 10.4.5

Therefore, we first consider the case when ξ has a discrete distribution. Secondly, we introduce a method to approximate joint chance constraints by single chance constraints.

2.4.1 Finite discrete distribution

Literature: Kall and Mayer 2010, section 2.2.2

We assume that ξ has a finite discrete distribution. That is, it can attain only finitely many values $\xi_i, i \in I = \{1, ..., N\}$ with corresponding positive probabilities $p_i, i \in I$, $\sum_{i \in I} p_i = 1$.

Then for a given point x it holds

$$\mathbb{P}(T(\xi)x \geq h(\xi)) \geq \alpha \iff x \in \bigcup_{\substack{J \subset I \\ \sum_{j \in J} p_j \geq \alpha}} \bigcap_{j \in J} \{x \mid T(\xi_j)x \geq h(\xi_j)\}.$$

This means that x is feasible for the joint chance constraint if and only if there is a set of indices J such that x satisfies the constraints in all scenarios in J and the probability of this set of scenarios is at least α.

We observe that the sets

$$\bigcap_{j \in J} \{x \mid T(\xi_j)x \geq h(\xi_j)\}$$

are convex as intersections of convex polyhedral sets. However, the union of convex sets is in general not convex and consequently the feasible set for x is in general not convex. This is illustrated in Figure 2.10.

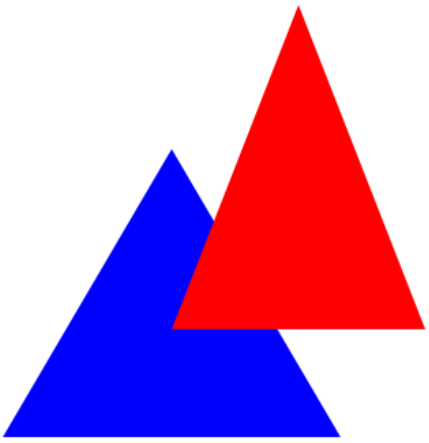

Figure 2.10: The feasible set is the union of the blue and the red polyhedron. This is not a convex set.

Problems of minimizing over a union of convex polyhedral sets are called disjunctive programming problems. They can be converted into mixed-integer linear programs. Therefore, for each scenario a binary variable $z_i, i \in I$ is introduced that indicates whether the constraints of this scenario are satisfied or not. This leads to the following reformulation of (JCCP):

$$\begin{aligned}
\text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\
\text{subject to} \quad & Ax = b \\
& x \geq 0 \\
& T(\xi_i)x + M(1 - z_i)\mathbb{1} \geq h(\xi_i)\,, \quad i \in I \\
& \sum_{i=1}^{N} p_i z_i \geq \alpha \\
& z_i \in \{0, 1\}\,, \quad i \in I
\end{aligned}$$

where $\mathbb{1} = (1, ..., 1)^\top$ is the vector of all ones and M is a "big enough" constant. To be more precise, M is chosen such that

$$M \cdot \mathbb{1} \geq h(\xi_i) - T(\xi_i)x \quad \forall x \in \{x \mid Ax = b, x \geq 0\}\ \forall i \in I.$$

If a scenario ξ_i is "active", that is $z_i = 1$, then

$$T(\xi_i)x + \underbrace{M \cdot (1 - z_i)\mathbb{1}}_{=0} \geq h(\xi_i)$$

and the original constraint for scenario i holds. On the other hand, if the scenario i is "inactive", that is $z_i = 0$, then

$$T(\xi_i)x + \underbrace{M \cdot (1 - z_i)}_{=M}\,\mathbb{1} \geq h(\xi_i)$$

holds independently of the choice of x. This means that the original constraint for this scenario is deactivated.

M needn't be chosen minimal. If possible, one can choose a value for M by some problem specific knowledge. Otherwise M can be computed as follows:

1. For $i \in I$ and $j \in \{1, ..., s\}$ let M_{ij} be the optimal value of the linear program

$$\begin{aligned} \text{maximize} \quad & h_j(\xi_i) - t_j(\xi_i)x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

   where $t_j(\xi_i)$ denotes the j$^{\text{th}}$ row of $T(\xi_i)$ and $h_j(\xi_i)$ denotes the j$^{\text{th}}$ entry of $h(\xi_i)$.

2. Setting

$$M := \max_{\substack{i \in I \\ j \in \{i=1,...,s\}}} M_{ij}$$

   provides a possible constant M.

Even though linear programs are efficiently solvable, this approach requires to solve $N \cdot s$ many linear programs which can be computationally expensive for a huge number of scenarios N and uncertain constraints s.

2.4.2 Approximation by discretization

Bearing in mind the previous section an obvious approach to approximate joint chance constraints with an arbitrary distribution is by discretization. That is, a finite number of random samples ξ_i, $i = 1, ..., N$ of the distribution is drawn. Then the mixed-integer reformulation of the previous section can be used. This is also called sample average approximation. For further information see Geng and Xi 2019, chapter 6.

Another - more conservative - approach is to require that the uncertain constraints have to hold on all the randomly drawn samples. This is also called the scenario approach. This leads to the problem

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \\ & T(\xi_i)x \geq h(\xi_i) \quad \forall i \in I. \end{aligned}$$

We observe that this is a linear program and hence way easier to solve than the mixed-integer linear program before. For further information see Geng and Xi 2019, chapter 5.

Remark 2.31
Both approaches do not provide deterministic guarantees. It may happen that an optimal solution to one of the approaches is not even feasible for the original problem. However, one can show that such a solution is feasible for the original problem with a certain probability (depending on the number of samples N and increasing for bigger N).

2.4.3 Approximation by single chance constraints

As we saw in the previous section, discrete approximation strategies ensure feasibility of solutions only with a certain probability. Contrary to that, in this section we present a way to ensure feasibility in all cases. This approach makes use of single chance constraints.

Theorem 2.32 (Safe approximation of joint chance constraints)
The feasible set of the single chance constraints

$$\mathbb{P}(t_i(\xi)x \geq h_i(\xi)) \geq 1 - \epsilon_i\,, \quad i = 1, ..., s \tag{2.45}$$

with

$$\sum_{i=1}^{s} \epsilon_i \leq \epsilon$$

is a subset of the feasible set of the joint chance constraint

$$\mathbb{P}(T(\xi)x \geq h(\xi)) \geq 1 - \epsilon. \tag{2.46}$$

Proof. We show that if x is feasible for (2.45), then it is also feasible for (2.46). Using the union bound for probabilities leads to

$$\begin{aligned}
\mathbb{P}(T(\xi)x \geq h(\xi)) &= \mathbb{P}\Big(\bigcap_{i=1}^{s} \{t_i(\xi)x \geq h_i(\xi)\}\Big) \\
&= 1 - \mathbb{P}\Big(\bigcup_{i=1}^{s} \{t_i(\xi)x < h_i(\xi)\}\Big) \\
&\geq 1 - \sum_{i=1}^{s} \underbrace{\mathbb{P}(t_i(\xi)x < h_i(\xi))}_{\leq \epsilon_i} \\
&\geq 1 - \epsilon
\end{aligned}$$

□

To be more specific, for $\sum_{i=1}^{s} \epsilon_i \leq \epsilon$ the optimal solution of the single chance constraint problem

$$\begin{aligned}
\text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\
\text{subject to} \quad & Ax = b \\
& x \geq 0 \\
& \mathbb{P}(t_i(\xi)^\top x \geq h_i(\xi)) \geq 1 - \epsilon_i \quad \text{for } i = 1, ..., s
\end{aligned} \tag{2.47}$$

is also feasible (though not necessarily optimal) for the joint chance constraint problem

$$\begin{aligned}
\text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\
\text{subject to} \quad & Ax = b \\
& x \geq 0 \\
& \mathbb{P}(T(\xi)x \geq h(\xi)) \geq 1 - \epsilon.
\end{aligned}$$

This is also called a safe approximation. A typical choice for ϵ_i is $\epsilon_i = \frac{\epsilon}{s}$ for all i and hence the constraints

$$\mathbb{P}(t_i(\xi)x \geq h_i(\xi)) \geq 1 - \frac{\epsilon}{s} \quad \text{for } i = 1, ..., s.$$

It is quite difficult to determine good values for the ϵ_i. For further information see Geng and Xi 2019, section 7.4.1.

If the single chance constraints are multivariate normal, then the techniques of Section 2.2 can be used to solve (2.47). Otherwise approximation strategies for single chance constraints can be applied (compare Remark 2.5).

2.4.4 Example: Production planning of wind energy

An example where general joint chance constraints occur are renewable energies, for instance wind energy. Here the joint chance constraint

$$\mathbb{P}(T(\xi)x \geq h(\xi)) \geq \alpha$$

has the following interpretation:

1. The right hand side vector $h(\xi)$ represents varying electrical power demands of the customers.
2. The matrix $T(\xi)$ represents uncertainties in the production process where wind is considered as a random influence.

The joint chance constraint ensures that the random demands are satisfied with a probability of at least α while incorporating also the randomness in the production process. For details see Geng and Xi 2019, section 8.1.2.

3 Two-stage models

In this chapter we want to solve the optimization problem

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & T(\xi)x = h(\xi) \\ & x \geq 0. \end{aligned} \tag{3.1}$$

Here $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ are assumed to be deterministic. $T(\xi) \in \mathbb{R}^{s \times n}$ is a random matrix and $h(\xi) \in \mathbb{R}^s$ a random vector, both depending on the random vector $\xi \in \mathbb{R}^r$ with known probability distribution. We assume that the distribution of ξ does not depend on x.

As (3.1) contains unknown parameters, we first have to assign a formal meaning to it. In situations like the presence of a legal commitment it is important that the random constraints always hold. Therefore, in this chapter we consider an approach where a recourse action is taken to ensure the satisfaction of the uncertain constraints.

1. We make a first stage decision x as the optimal solution of the first stage problem

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{3.2}$$

2. Then the realization of the randomness ξ occurs.

3. Afterwards, we make a second stage decision y to compensate for the deficiency $h(\xi) - T(\xi)x$ with minimal cost. Therefore, we solve the problem

$$\begin{aligned} \text{minimize} \quad & q(\xi)^\top y \quad \text{where } y \in \mathbb{R}^k \\ \text{subject to} \quad & W(\xi)y = h(\xi) - T(\xi)x \\ & y \geq 0 \end{aligned} \tag{3.3}$$

   for the fixed first stage solution x and the fixed realization of the random vector ξ. Here are $W \in \mathbb{R}^{s \times k}$, $q \in \mathbb{R}^k$. We denote the optimal value of (3.3) by $Q(\xi, x)$.

Two-stage models are used if a problem is solved many times, for instance in weekly production planning. In these situations it is reasonable to consider the average recourse cost at a given first stage solution x, i.e.

$$\mathbb{E}[Q(\xi, x)],$$

as the metric to determine the quality of a first stage solution with respect to the uncertain constraints $T(\xi)x = h(\xi)$. Accordingly, in this chapter we consider the following problem:

$$\begin{aligned} \text{minimize} \quad & c^\top x + \mathbb{E}[Q(\xi, x)] \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{3.4}$$

In this chapter we mainly deal with the case when ξ has a finite discrete distribution. We briefly consider the case of general distributions in Section 3.3.

3.1 Finite discrete distribution

In this section we assume that the random vector ξ has only finitely many realizations $\xi_i, i = 1, ..., N$ with corresponding probabilities $p_i, i = 1, ..., N$, $\sum_{i=1}^{N} p_i = 1$. Then we can write problem (3.4) as

$$\begin{aligned} \text{minimize} \quad & c^\top x + \sum_{i=1}^{N} p_i q(\xi_i)^\top y_i \quad \text{where } x \in \mathbb{R}^n, y_i \in \mathbb{R}^{k_i}, i = 1, ..., N \\ \text{subject to} \quad & Ax = b \\ & T(\xi_i)x + W(\xi_i)y_i = h(\xi_i), \ i = 1, ..., N \\ & x \geq 0 \\ & y_i \geq 0, \ i = 1, ...N, \end{aligned} \tag{3.5}$$

where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ and $T(\xi_i) \in \mathbb{R}^{s_i \times n}, W(\xi_i) \in \mathbb{R}^{s_i \times k_i}, h(\xi_i) \in \mathbb{R}^{s_i}, q(\xi_i) \in \mathbb{R}^{k_i}$ for $i = 1, ..., N$.

We observe that (3.5) is again a linear program and the constraint matrix (without the nonnegativity constraints) has the following structure:

$$\begin{pmatrix} A & & & & \\ T(\xi_1) & W(\xi_1) & & & \\ T(\xi_2) & & W(\xi_2) & & \\ \vdots & & & \ddots & \\ T(\xi_N) & & & & W(\xi_N) \end{pmatrix} \tag{3.6}$$

For many scenarios (i.e. big N) this matrix might be very large and and hence (3.5) is a large-scale linear program. However, this matrix has a block structure that can be used in solution methods. The main observation is the following: If x was fixed, then (3.5) would decompose into the scenarios $i = 1, ..., N$ and we could solve the subproblems

$$\begin{aligned} \text{minimize} \quad & q(\xi_i)^\top y_i \quad \text{where } y_i \in \mathbb{R}^{k_i} \\ \text{subject to} \quad & T(\xi_i)x + W(\xi_i)y_i = h(\xi_i) \\ & y_i \geq 0 \end{aligned} \tag{3.7}$$

independently of each other. As the complexity of solving linear programs grows more than linear, this would be a big speed-up in computation time. Using this observation is the main ingredient of a method called Benders decomposition which we introduce now.

3.2 Benders decomposition

3.2.1 Motivation of the method

Literature: Taskin 2010, section 1

To increase readability we derive the method for the case $N = 1$. It generalizes straightforwardly to arbitrary N. We cover this in Section 3.2.4. We consider the following problem:

$$\begin{aligned} \text{minimize} \quad & c^\top x + q^\top y \quad \text{where } x \in \mathbb{R}^n,\ y \in \mathbb{R}^k \\ \text{subject to} \quad & Ax = b \\ & Tx + Wy = h \\ & x \geq 0 \\ & y \geq 0. \end{aligned} \tag{3.8}$$

We assume that it is solvable. Additionally, we assume that the feasible set of the first stage problem

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{3.9}$$

is bounded (which implies also solvability of (3.9)). For a given vector x (e.g. an optimal first stage solution), the second stage problem reads

$$\begin{aligned} \text{minimize} \quad & q^\top y \quad \text{where } y \in \mathbb{R}^k \\ \text{subject to} \quad & Wy = h - Tx \\ & y \geq 0. \end{aligned} \tag{3.10}$$

Its dual problem is

$$\begin{aligned} \text{maximize} \quad & (h - Tx)^\top u \quad \text{where } u \in \mathbb{R}^s \\ \text{subject to} \quad & W^\top u \leq q. \end{aligned} \tag{3.11}$$

The solvability of (3.8) implies that the feasible set of (3.11) is nonempty. A nonempty polyhedron has a finite number of extreme points/vertices and extreme rays (see Nemhauser and Wolsey 1999, section I.4.4). We denote the set of vertices by V and the set of extreme rays by R. Then a linear program over this nonempty polyhedron can be solved by first checking that the objective does not increase along the extreme rays to ensure boundedness and if so selecting an optimal vertex. Due to linear programming duality (3.11) has the same optimal value as

$$\begin{aligned} \text{minimize} \quad & \theta \quad \text{where } \theta \in \mathbb{R} \\ \text{subject to} \quad & (h - Tx)^\top r \leq 0 \quad \forall r \in R \\ & (h - Tx)^\top v \leq \theta \quad \forall v \in V. \end{aligned} \tag{3.12}$$

We note that if $(h - Tx)^\top r > 0$ for some $r \in R$, then the feasible set of (3.12) is empty and its optimal value is ∞. In this case (3.11) is unbounded and hence its optimal value is also ∞.

Consequently, we can write (3.8) as

$$\begin{aligned} \text{minimize} \quad & c^\top x + \theta \quad \text{where } x \in \mathbb{R}^n,\ \theta \in \mathbb{R} \\ \text{subject to} \quad & Ax = b \\ & (h - Tx)^\top r \le 0 \quad \forall r \in R \\ & (h - Tx)^\top v \le \theta \quad \forall v \in V \\ & x \ge 0. \end{aligned} \tag{3.13}$$

However, the size of V and R grows exponentially with the problem size. Hence, this formulation is not directly useful to solve problem (3.8). Therefore, Benders decomposition uses only a subset of this constraints and sequentially adds more constraints if they turn out to be necessary along the method. In each iteration Benders decomposition fixes a value of x by solving a master problem and then solves the dual problem (3.11) of the second stage problem. Based on the solution of the dual problem a new constraint is generated.

3.2.2 Formal derivation of the method

Literature: Kall and Mayer 2010, section 1.2.6

Now we formally derive the Benders decomposition method. It is summarized in Algorithm 2.

In iteration $k = 0$ we only use the first stage constraints and the feasible set reads

$$P_0 := \{(x, \theta) \mid Ax = b, x \ge 0\}.$$

Then we solve the first stage problem

$$\begin{aligned} \text{minimize} \quad & c^\top x + \theta \quad \text{where } x \in \mathbb{R}^n,\ \theta \in \mathbb{R} \\ \text{subject to} \quad & x \in P_k \end{aligned} \tag{3.14}$$

yielding the optimal solution (x_k^*, θ_k^*). If there is no constraint for θ available in P_k (as it is the case in $k = 0$), then θ is ignored in the objective and we set $\theta_k^* = -\infty$. We note that the solvability of (3.8) implies the feasibility of (3.14). Moreover, the bounded feasible set of (3.9) implies also the boundedness of (3.14) since: Assume that (3.14) is unbounded. This means there is an extreme ray $(\Delta x, \Delta\theta)$ with negative objective value for (3.14), i.e.

$$A\Delta x = 0, \Delta x \ge 0,$$

$$\begin{aligned} -(T\Delta x)^\top r \le 0 \quad & \forall r \in \hat{R}, \\ -(T\Delta x)^\top v \le \Delta\theta \quad & \forall v \in \hat{V}, \end{aligned} \tag{3.15a}$$

and

$$c^\top \Delta x + \Delta\theta < 0. \tag{3.15b}$$

Here $\hat{R} \subset R$ and $\hat{V} \subset V$ correspond to the constraints that have been added to (3.14) until iteration k. As we assumed boundedness of the feasible set of (3.9), it holds $\Delta x = 0$. Then (3.15b) implies $\Delta\theta < 0$. But this contradicts (3.15a). Hence, (3.14) is bounded.

With the optimal first stage solution x_k^* we solve the dual of the second stage problem, i.e.

$$\begin{aligned} &\text{maximize} && (h - Tx_k^*)^\top u \quad \text{where } u \in \mathbb{R}^s \\ &\text{subject to} && W^\top u \leq q. \end{aligned} \tag{3.16}$$

We note that infeasibility cannot occur due to the solvability of (3.8). Accordingly, we distinguish two cases regarding to the optimal value of (3.16).

1. If (3.16) is unbounded, then the primal second stage problem

$$\begin{aligned} &\text{minimize} && q^\top y \quad \text{where } y \in \mathbb{R}^k \\ &\text{subject to} && Wy = h - Tx_k^* \\ & && y \geq 0 \end{aligned} \tag{3.17}$$

   is infeasible. This means that for the first stage solution x_k^* there is no $y \geq 0$ such that

$$Wy = h - Tx_k^*.$$

   Hence, x_k^* is infeasible for (3.8) and should be cut off. As (3.16) was unbounded, the simplex method detects a feasible direction r of unbounded growth (an extreme ray), i.e.

$$W^\top r \leq 0$$

   and

$$(h - Tx_k^*)^\top r > 0.$$

   For any feasible x for (3.8) there is a $y \geq 0$ such that $Wy = h - Tx$ and hence

$$(h - Tx)^\top r = y^\top (W^\top r) \leq 0.$$

   Thus, the constraint

$$(h - Tx)^\top r \leq 0$$

   cuts off the infeasible point x_k^*. This is called a feasibility cut. We observe that this is one of the constraints of the formulation (3.13). We set

$$P_{k+1} \coloneqq P_k \cap \{(x, \theta) \mid (h - Tx)^\top r \leq 0\}$$

   and solve (3.14) again.

2. If (3.16) turns out to be solvable with an optimal vertex v and optimal value $f_k^* \coloneqq (h - Tx_k^*)^\top v$, we check whether

$$f_k^* = \theta_k^*$$

   holds. We note that $f_k^* \geq \theta_k^*$ always holds.

If so, then x_k^* and the corresponding primal solution y_k^* of (3.16) are optimal for (3.8) since: (3.14) is a relaxation of (3.13) (it simply has less constraints). Hence, if (x_k^*, θ_k^*) is feasible for (3.13), then it is also optimal. The finiteness of f_k^* implies that the constraints for the extreme rays $r \in R$ in (3.13) hold. Moreover it implies that no vertex of the polyhedron of the dual problem (3.16) has an objective value greater than f_k^*. As $f_k^* = \theta_k^*$, (x_k^*, θ_k^*) satisfies also the constraints for the vertices in (3.13). This shows that (x_k^*, θ_k^*) is also feasible for (3.13) and hence optimal.

In any case,

$$\theta \geq (h - Tx)^\top v$$

has to hold. This is one of the constraints in (3.13). It holds $f_k^* = (h - Tx^*)^\top v$. But for $f_k^* > \theta_k^*$ this constraint is violated for (x_k^*, θ_k^*). Therefore we set

$$P_{k+1} \coloneqq P_k \cap \{(x, \theta) \mid \theta \geq (h - Tx)^\top v\}$$

and cut off (x_k^*, θ_k^*). This is called an optimality cut. Then we solve (3.14) again.

Algorithm 2 Benders decomposition

1: **Input**: A problem of type (3.8).
2: **Output**: Optimal solution (x^*, y^*) for (3.8).
3: **Initialization**: Set $k = 0$, $\theta_0^* := -\infty$, $P_0 := \{(x, \theta) \mid Ax = b, x \geq 0\}$ and solve

$$\begin{aligned} &\text{minimize} && c^\top x \quad \text{where } x \in \mathbb{R}^n \\ &\text{subject to} && x \in P_0 \end{aligned}$$

yielding the optimal solution x_0^*.
4: Solve the dual second stage problem

$$\begin{aligned} &\text{maximize} && (h - Tx_k^*)^\top u \quad \text{where } u \in \mathbb{R}^s \\ &\text{subject to} && W^\top u \leq q \end{aligned} \tag{3.18}$$

and let f_k^* be its optimal value.
5: **if** $f_k^* = \infty$ **then**
6: The dual simplex method determined an unbounded growth direction r.
7: Set $P_{k+1} := P_k \cap \{(x, \theta) \mid (h - Tx)^\top r \leq 0\}$ (**feasibility cut**)
8: **else**
9: There is an optimal vertex v such that $f_k^* = (h - Tx_k^*)^\top v$.
10: **if** $f_k^* = \theta_k^*$ **then**
11: return x_k^* and the corresponding primal solution y_k^* to (3.18).
12: **else**
13: Set $P_{k+1} := P_k \cap \{(x, \theta) \mid \theta \geq (h - Tx)^\top v\}$ (**optimality cut**)
14: **end if**
15: **end if**
16: Solve

$$\begin{aligned} &\text{minimize} && c^\top x + \theta \quad \text{where } x \in \mathbb{R}^n,\ \theta \in \mathbb{R} \\ &\text{subject to} && x \in P_{k+1} \end{aligned}$$

yielding optimal solutions $(x_{k+1}^*, \theta_{k+1}^*)$. θ is neglected in the objective as long as there is no constraint on θ and is set to $\theta_{k+1}^* = -\infty$.
17: $k := k + 1$, GOTO 3

Remark 3.1 (Solvability assumptions)
If the assumptions on (3.8) and (3.9) are not satisfied, the following holds:

1. *If the first stage problem (3.9) is not solvable, this is detected in line 2. However, it might happen that (3.9) is solvable but the feasible set is not bounded. If the feasible set of (3.9) is not bounded, then it may happen that the relaxed problem (3.14) is unbounded in some iteration although (3.13) is solvable. In this case, the method fails.*

2. *If the original problem (3.8) is not solvable, this is detected either by the infeasibility of the problem in line 3, or by infeasibility or unboundedness of the problem in line 15.*

3.2.3 Convergence

Theorem 3.2 (Convergence of Benders decomposition)
Provided that (3.8) is solvable and the feasible set of (3.9) is bounded, Algorithm 2 yields an optimal solution (x^*, y^*) *to (3.8) after finitely many iterations.*

Proof. As a polyhedron has only finitely many extreme rays and extreme points and in each iteration one of the missing constraints of (3.13) is added, the algorithm stops with an optimal solution after finitely many iterations. □

Of course, finite convergence is a rather weak statement as the number of extreme points and extreme rays of a polyhedron can grow exponentially with the problem size and hence in the worst case the number of iterations grows exponentially with the problem size. However, similar to the simplex algorithm which also has a exponential worst case runtime, in practice Benders decomposition works pretty well.

To illustrate this, I implemented the Benders decomposition method. The arising linear programming subproblems in Algorithm 2 were solved with the general purpose linear programming solver Gurobi. As a comparison, randomized two-stage models with a varying number of scenarios N were solved in two ways:

1. With my own implementation of Benders decomposition where Gurobi is only used to solve the subproblems.

2. Directly with Gurobi. That is, the typically large-scale matrix (3.6) was directly passed as an argument and the block structure was not used.

The result is visualized in Figure 3.1. We can see that for a small number of scenarios Gurobi works still better as my own implementation. However, as the number of scenarios increases Benders decomposition performs way better than Gurobi.

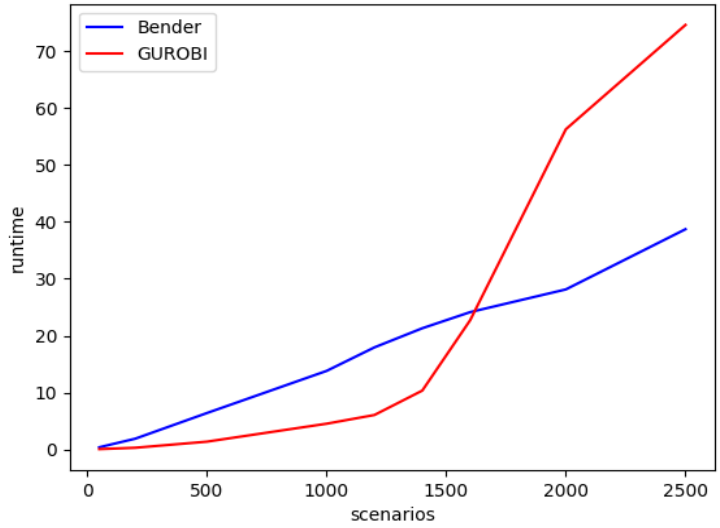

Figure 3.1: Numerical comparison of solving two-stage models with Benders decomposition and the general purpose linear programming solver Gurobi.

Moreover, after iteration k the optimal value $c^\top x_k^* + \theta_k^*$ of (3.14) provides a lower bound for (3.8) whereas $c^\top x_k^* + q^\top y_k^*$ yields an upper bound for y_k^* being the corresponding primal optimal solution of (3.16) (provided there is one).

3.2.4 Application to two-stage models

Literature: Prékopa 1995, section 12.4

Now we apply Benders decomposition to an arbitrary number of scenarios N. The resulting method is also called the L-shaped method. It is an immediate generalization of the method we introduced in the previous section.

Therefore we denote the vertices and extreme rays of the dual feasible sets $\{u_i \mid W(\xi_i)^\top u_i \leq q(\xi_i)\}, i = 1, ..., N$ by V_i and R_i, $i = 1, ..., N$, respectively.

Similar as in (3.13) we can write the general discrete two-stage problem (3.5) as

$$\begin{aligned} \text{minimize} \quad & c^\top x + \theta \quad \text{where } x \in \mathbb{R}^n,\ \theta \in \mathbb{R} \\ \text{subject to} \quad & Ax = b \\ & (h(\xi_i) - T(\xi_i)x)^\top r \leq 0 \quad \forall r \in R_i\ \forall i = 1, ..., N \\ & \sum_{i=1}^{N} p_i \cdot (h(\xi_i) - T(\xi_i)x)^\top v_i \leq \theta \quad \forall v_i \in V_i\ \forall i = 1, ..., N \\ & x \geq 0. \end{aligned} \tag{3.19}$$

Under the assumption that (3.19) is solvable and the first stage problem

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

has a bounded feasible set, Algorithm 2 generalizes to Algorithm 3.

Algorithm 3 L-shaped method

1: **Input**: A problem of type (3.5)
2: **Output**: Optimal solution $(x^*, y_1^*, ..., y_N^*)$ for (3.5).
3: **Initialization**: Set $k = 0$, $\theta_0^* := -\infty$, $P_0 := \{(x, \theta) \mid Ax = b, x \geq 0\}$ and solve

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & x \in P_0 \end{aligned} \tag{3.20}$$

yielding the optimal solution x_0^*.
4: For $i = 1, ..., N$ solve the dual second stage problem

$$\begin{aligned} \text{maximize} \quad & (h(\xi_i) - T(\xi_i)x_k^*)^\top u_i \quad \text{where } u_i \in \mathbb{R}^{s_i} \\ \text{subject to} \quad & W(\xi_i)^\top u_i \leq q(\xi_i) \end{aligned} \tag{3.21}$$

and let f_{ki}^* be its optimal value.
5: **if** $f_{ki}^* = \infty$ for some $i = 1, ..., N$ **then**
6: Let i be the first index such that $f_{ki}^* = \infty$.
7: The dual simplex method determined an unbounded growth direction r for subproblem i.
8: Set $P_{k+1} := P_k \cap \{(x, \theta) \mid (h(\xi_i) - T(\xi_i)x)^\top r \leq 0\}$ (**feasibility cut**)
9: **else**
10: There is an optimal vertex v_i such that $f_{ki}^* = (h(\xi_i) - T(\xi_i)x_k^*)^\top v_i \quad \forall i = 1, ..., N$.
11: **if** $\sum_{i=1}^N p_i \cdot (h(\xi_i) - T(\xi_i)x)^\top v_i = \theta_k^*$ **then**
12: return x_k^* and the corresponding primal optimal solutions y_{ki}^*, $i = 1, ..., N$ of (3.21)
13: **else**
14: Set $P_{k+1} := P_k \cap \{(x, \theta) \mid \theta \geq \sum_{i=1}^N p_i \cdot (h(\xi_i) - T(\xi_i)x)^\top v_i\}$ (**optimality cut**)
15: **end if**
16: **end if**
17: Solve

$$\begin{aligned} \text{minimize} \quad & c^\top x + \theta \quad \text{where } x \in \mathbb{R}^n,\ \theta \in \mathbb{R} \\ \text{subject to} \quad & x \in P_{k+1} \end{aligned} \tag{3.22}$$

yielding optimal solutions $(x_{k+1}^*, \theta_{k+1}^*)$. θ is neglected in the objective as long as there is no constraint on θ and is set to $\theta_{k+1}^* = -\infty$.
18: $k := k + 1$, GOTO 3

Remark 3.3

1. *The only difference between Algorithm 3 and Algorithm 2 is that in each step N subproblems*

are considered instead of only one. In particular, an optimality cut is added in line 13 only if all subproblems have finite optimal values.

2. *The same convergence results as for $N = 1$ hold: The algorithm terminates with an optimal solution for (3.5) (provided there is one) after a finite number of iterations as there are finitely many subproblems and each subproblem has only a finite number of vertices and extreme rays. In practice the convergence is typically quite fast.*
   Again $c^\top x_k^* + \theta_k^*$ is available as a lower bound in iteration k whereas $c^\top x_k^* + \sum_{i=1}^N p_i \cdot q_i^\top y_{ki}^*$ yields an upper bound for y_{ki}^* being the optimal primal solutions (provided they exist) for the dual problems (3.21).

3.3 The general case

In this section we consider the general two-stage model (3.4), i.e.

$$\begin{aligned} \text{minimize} \quad & c^\top x \; + \; \mathbb{E}[Q(\xi, x)] \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where ξ has an arbitrary probability distribution. We only give a rough overview over an approximation procedure. For further details see Kall and Mayer 2010, section 3.2.1.

Solution methods for those problems are based on successive discretization strategies. Under some extra assumptions (e.g. that W is deterministic) convexity properties for $\mathbb{E}[Q(\xi, x)]$ can be proven. They can be used to construct two-stage models with finite discrete distributions that lead to lower and upper bounds for (3.4). Jensens inequality is used to get a lower bound L, i.e.

$$L(x) \leq \mathbb{E}[Q(\xi, x)],$$

and the so called Edmundson-Madansky inequality leads to an upper bound U, i.e.

$$\mathbb{E}[Q(\xi, x)] \leq U(x).$$

In each iteration two-stage models with finite discrete distribution are solved. This can be done by the L-shaped method we presented in the previous section. Moreover, refining the discretization leads to increasing lower and decreasing upper bounds. Moreover one can show that L and U converge to $\mathbb{E}[Q(\xi, x)]$ as the discretization gets finer.

3.4 Example: Transport problem with random demands revisited

Similar to Section 2.3.3 we consider a transport problem of the form

$$\begin{aligned} \text{minimize} \quad & c^\top x \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & A_{\text{supply}} x = b_{\text{supply}} \\ & A_{\text{demand}} x = b_{\text{demand}}(\xi) \\ & 0 \leq x \leq u. \end{aligned}$$

We again assume that the customer demands are random. However, contrary to Section 2.3.3 we now want to satisfy the customer demands in any case. Therefore, we formulate the corresponding two-stage problem

$$\begin{aligned} \text{minimize} \quad & c^\top x + \mathbb{E}[Q(\xi, x)] \quad \text{where } x \in \mathbb{R}^n \\ \text{subject to} \quad & A_{\text{supply}} x = b_{\text{supply}} \\ & 0 \le x \le u. \end{aligned} \tag{3.23}$$

where $Q(\xi, x)$ is the optimal value of

$$\begin{aligned} \text{minimize} \quad & q(\xi)^\top y \quad \text{where } y \in \mathbb{R}^k \\ \text{subject to} \quad & W(\xi) y = b_{\text{demand}}(\xi) - A_{\text{demand}} x \\ & y \ge 0 \end{aligned}$$

for fixed ξ and x.

We can interpret W as the structure of an emergency transport network. The first stage network given by the matrix $A = (A_{\text{supply}}, A_{\text{demand}})$ may describe the structure of a ship network. As ships need a lot of time to cover long distances, there is a big time delay between the start of the transportation and the arrival at the customers. Therefore, W may describe a transport network consisting of warehouses to compensate for an excess of the commodity, and airplanes to quickly compensate for a deficiency of the commodity. Evidently, warehouses as well as airplanes are way more flexible than ships. However, they also entail costs $q(\xi)$ that might be random as well as the structure of the recourse network $W(\xi)$ itself. Therefore, (3.23) finds a balance between the flexibility of the recourse network and the rather small costs of ship transport compared to airplane transport.

Bibliography

Alizadeh, F. and D. Goldfarb (2002). "Second-order cone programming". In: *Mathematical Programing, Series B 95, 3–51 (2003)*. DOI: 10.1007/s10107-002-0339-5.

Bazaraa, M. S., H. D. Sherali, and C. Shetty (2006). *Nonlinear programming, theory and algorithms*. John Wiley Sons, Inc. DOI: 10.1002/0471787779.

Ben-Tal, A., L. E. Ghaoui, and A. Nemirovski (2009). *Robust Optimization*. Princeton University Press. DOI: 10.1515/9781400831050.

d'Aspremont, A. (2023). *Convex Optimization M2 Lecture 3*. URL: https://www.di.ens.fr/~aspremon/PDF/MVA/Duality.pdf.

Geiger, C. and C. Kanzow (2002). *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer Berlin, Heidelberg. DOI: 10.1007/978-3-642-56004-0.

Geng, X. and L. Xi (2019). "Data-driven decision making in power systems with probabilistic guarantees: Theory and applications of chance-constrained optimization". In: *Annual Reviews in Control*. DOI: 10.1016/j.arcontrol.2019.05.005.

Herzog, R. (2023). *Vorlesungsskript Grundlagen der Optimierung*. URL: https://scoop.iwr.uni-heidelberg.de/teaching/2022ws/lecture-grundlagen-der-optimierung/grundlagen-der-optimierung-skript-20230207.pdf.

Kall, P. and J. Mayer (2010). *Stochastic Linear Programming*. Springer New York, NY. DOI: 10.1007/978-1-4419-7729-8.

Kummer, B. (2011). *Counterexamples in Optimization and non-smooth Analysis*. URL: https://www.mathematik.hu-berlin.de/~kummer/scripts/gegenBei.pdf.

Lobo, M. S., L. Vandenberghe, S. Boyd, and H. Lebret (1998). "Applications of second-order cone programming". In: *Linear Algebra and its Applications*. DOI: 10.1016/S0024-3795(98)10032-0.

Nemhauser, G. and L. Wolsey (1999). *Integer and Combinatorial Optimization*. John Wiley Sons Inc. DOI: 10.1002/9781118627372.

Nocedal, J. and S. J. Wright (2006). *Numerical Optimization*. Springer New York, NY. DOI: 10.1007/b98874.

Prékopa, A. (1995). *Stochastic Programming*. Springer Dordrecht. DOI: 10.1007/978-94-017-3087-7.

Szántai, T. (1988). "A computer code for solution of probabilistic-constrained stochastic programming problems". In: *Numerical techniques for stochastic optimization*, pp. 229–235.

Taskin, Z. C. (2010). "Benders decomposition". In: *Wiley Encyclopedia of Operations Research and Management Science. John Wiley Sons, Malden (MA)*. DOI: 10.1002/9780470400531.eorms0104.

Veinott, A. F. (1967). "The Supporting Hyperplane Method for Unimodal Programming". In: *Operations Research* 15.1, pp. 147–152. DOI: 10.1287/opre.15.1.147.

Declaration

I hereby confirm that I wrote this work independently and did not use any sources other than those indicated.

Heidelberg, July 13, 2023 .....

Max Jungmann