# A New Relaxation Technique for Polynomial Optimization and Spectrahedral Geometry Problems

Dissertation zur Erlangung des
Doktorgrades der Naturwissenschaften

vorgelegt beim Fachbereich 12
– Informatik und Mathematik –
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main


von Christian Trabandt aus Bad Nauheim

Frankfurt 2014
(D 30)

Vom Fachbereich 12 – Informatik und Mathematik – der
Johann Wolfgang Goethe - Universität als Dissertation angenommen.

| | |
|---|---|
| Dekan: | Prof. Dr. Thorsten Theobald |
| Gutachter: | Prof. Dr. Thorsten Theobald<br>PD Dr. Daniel Plaumann |
| Datum der Disputation: | 04.11.2014 |

# Zusammenfassung

Diese Arbeit beschäftigt sich mit zwei Themenkomplexen im Schnitt der Bereiche der konvexen algebraischen Geometrie und der nichtlinearen Optimierung.

Zunächst wird ein neues Verfahren zum Lösen polynomieller Optimierungsprobleme über Polytopen entwickelt. Aus Sicht der konvexen algebraischen Geometrie werden derartige Probleme üblicherweise mithilfe von Positivstellensätzen aus der reellen algebraischen Geometrie modelliert. Durch die Beschränkung der Grade der Positivstellensatzzertifikate erhält man so eine Hierarchie von semidefiniten Programmen, die jeweils untere Schranken für das ursprüngliche Problem liefern. In der Optimierung ist es gängig, nichtlineare Optimierungsprobleme mit Hilfe von Branch and Bound-Verfahren zu lösen. Wir entwickeln in dieser Arbeit ein neues Branch and Bound-Verfahren, bei dem die Positivstellensatzrelaxierung zum Einsatz kommt, um untere Schranken zu generieren. Darüber hinaus entwickeln wir eine neue Fehlerschranke für die Relaxierung, die auf Handelmans Positivstellensatz beruht. Dieses Ergebnis wird im Folgenden genutzt, um die Konvergenz des resultierenden Branch and Bound-Verfahrens zu beweisen.

Der zweite Schwerpunkt dieser Arbeit liegt auf Enthaltenseinsfragen für Polyeder und Spektraeder. Durch die Anwendung von Positivstellensätzen hat die semidefinite Programmierung in den letzten Jahren stark an Bedeutung für die Approximation polynomieller Optimierungsprobleme gewonnen. Während die Zulässigkeitsbereiche der linearen Programmierung, Polytope und Polyeder, bereits eingehend studiert und weitestgehend verstanden sind, sind die Zulässigkeitsbereiche der semidefiniten Programmierung, sogenannte Spektraeder, ein wichtiger Gegenstand aktueller Forschung. Wir klassifizieren zunächst die Komplexität verschiedener Enthaltenseinsfragen von Polyedern und Spektraedern. Darüber hinaus wird eine Hierarchie von hinreichenden Bedingungen für das Enthaltensein eines Spektraeders in einem zweiten vorgestellt.

Im folgenden wird auf beide Themenbereiche detailliert eingegangen.

### Positivstellensatz Branch and Bound-Verfahren

Wir entwickeln ein neues Verfahren zum Lösen von polynomiellen Optimierungsproblemen, das heißt der Optimierung einer polynomiellen Zielfunktion $p(x) \in \mathbb{R}[x_1, \ldots, x_n]$ über einer semialgebraischen Menge $K$,

$$\begin{aligned} \text{minimiere} \quad & p(x) \\ \text{unter der Nebenbed.} \quad & x \in K. \end{aligned}$$

Diese Klasse von Optimierungsproblemen ist sehr weit gefasst, es können unter anderem viele kombinatorische Optimierungsprobleme in der obigen Form formuliert werden. Entsprechend ist das Lösen dieser Probleme im Allgemeinen äußerst schwierig. In der konvexen algebraischen Geometrie ist es eine gängige Herangehensweise, mit Hilfe von Positivstellensätzen Hierarchien von semidefiniten Programmen aufzustellen, deren Lösungen Näherungen an die Optimallösung des polynomiellen Optimierungsproblems liefern. Beginnend mit der Arbeit von Lasserre [46] wurde inzwischen eine große Bandbreite an Verfahren und Ergebnissen zu diesem Thema entwickelt ([51, 53]). Grundsätzlich betrachtet man das Problem zunächst in der dualen Form

$$\begin{aligned} &\text{maximiere} && t \\ &\text{unter den Nebenbed.} && p(x) - t \geq 0, \\ & && x \in K. \end{aligned}$$

Die Nebenbedingungen werden dann durch schwächere, aber einfacher handhabbare Bedingungen ersetzt. Abhängig vom verwendeten Positivstellensatz kann das resultierende Problem mit Hilfe der linearen oder der semidefiniten Programmierung gelöst werden.

Unser Hauptaugenmerk in dieser Arbeit liegt auf dem Fall, dass $K$ ein Polytop ist. In dieser Situation können wir Handelmans Positivstellensatz anwenden. Er besagt, dass ein Polynom $p(x) \in \mathbb{R}[x]$ genau dann positiv auf einem Polytop $P = \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_k(x) \geq 0, \text{ alle linear}\}$ ist, wenn es eine Darstellung der folgenden Form hat

$$p(x) = \sum_{\alpha \in \mathbb{N}^n} \lambda_\alpha g_1^{\alpha_1}(x) \cdots g_k^{\alpha_k}(x), \quad \lambda_\alpha \geq 0.$$

Durch Beschränkung des Grades des rechten Terms erhält man ein lineares Programm, für dessen Lösung äußerst effiziente Methoden zur Verfügung stehen. Mit steigender Gradschranke werden die Näherungen immer genauer, die relaxierten Probleme jedoch auch immer komplexer und entsprechend schwerer zu lösen. Um diesem Problem zu begegnen schlagen wir vor, den Zulässigkeitsbereich $K$ aufzuteilen und die resultierenden Unterprobleme auf einer festen Relaxierungsstufe zu lösen. Das resultierende Verfahren vereint die guten Approximationseigenschaften der Positivstellenverfahren mit einem Branch and Bound-Verfahren.

Unser Hauptbeitrag in diesem Gebiet ist die Herleitung einer neuen Fehlerschranke für das Handelman-Verfahren. Wir verallgemeinern eine Schranke für die Optimierung eines Polynoms über dem Einheitswürfel, die auf De Klerk und Laurent zurückgeht [13]. Hierfür nutzen wir die Ähnlichkeit der Handelman-Darstellung auf Hyperrechtecken mit Bernstein Polynomen. Unter Verwendung früherer Resultate zu Bernstein Darstellungen von Garloff [18] zeigen wir folgendes Resultat. Wir bezeichnen mit $p_{min,K}$ das Minimum von $p$ auf $K$, mit $p^{(d)}_{Hand,K}$ die entsprechende untere Schranke, die sich aus der Handelman-Relaxierung mit Gradschranke $d$ ergibt.

**Satz 3.15.** *Sei $K$ ein Polytop, welches in dem Hyperrechteck $[\underline{a}, \underline{b}]$ enthalten ist. Das Minimum $p_{min,[a,b]}$ von $p \in \mathbb{R}[x_1, \ldots, x_n]$ über dem Hyperrechteck werde in $K$ angenommen.*

*Sei $m \geq deg(p)$, dann gilt für jede natürliche Zahl $d \geq mn$,*

$$p_{min,K} - p^{(dn)}_{Hand,K} \leq \Delta(d),$$

*wobei für $\|b - a\|_\infty < 1$,*

$$\Delta(d) \leq \vartheta_p \max(1, |a_1|^m, \ldots, |a_n|^m) \|b - a\|^2_\infty \sum_{\gamma \in [m]^n_0} \left( \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^{n} \frac{(\gamma_j - 1)^2}{d} \right)$$

*und für $\|b - a\|_\infty \geq 1$,*

$$\Delta(d) \leq \vartheta_p \max(1, |a_1|^m, \ldots, |a_n|^m) \|b - a\|^{mn}_\infty \sum_{\gamma \in [m]^n_0} \left( \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^{n} \frac{(\gamma_j - 1)^2}{d} \right),$$

*mit einer Konstanten $\vartheta_p$, die von den Koeffizienten von $p$ abhängt.*

Die Schranke wurde explizit so entwickelt, dass sie von der längsten Kante eines das Polytop umschließenden Hyperrechtecks abhängt. Dies ermöglicht die Konvergenz des Branch and Bound-Verfahrens, welches wir nun kurz beschreiben, zu zeigen.

Die Grundidee des Verfahrens besteht darin, dass die Grundmenge des Optimierungsproblems sukzessive weiter unterteilt wird. In der vorliegenden Arbeit wird stets diejenige Teilmenge unterteilt, deren umschließendes Hyperrechteck eine Kante maximaler Länge hat. Die Menge wird entlang der längsten Kante halbiert und es wird eine untere und eine obere Schranken an das Minimum auf der entsprechenden Menge berechnet. Zur Berechnung einer unteren Schranke im $j$-ten Schritt $p_{lb,P}(j)$ wird das Handelman-Verfahren verwendet, die obere Schranke $p_{ub,P}(j)$ ergibt sich durch Einsetzen eines Punktes aus der Teilmenge.

**Satz 4.7.** *Das oben beschriebene Handelman Branch and Bound-Verfahren konvergiert, d.h. die untere Schranke $p_{lb,P}(j)$ und die obere Schranke $p_{ub,P}(j)$ konvergieren gegen $p_{min,K}$.*

Neben den theoretischen Ergebnissen werden die diskutierten Methoden auf eine Reihe von zufällig generierten Beispielen angewendet. Die numerischen Ergebnisse zeigen, dass die Unterteilung des Zulässigkeitsbereichs in der Tat zu genaueren Approximationen des Optimalwertes führt, bereits eine einzelne Unterteilung führt schon zu Verbesserungen.

## Enthaltenseinsfragen für Spektraeder

Enthaltenseinsfragen für konvexe Mengen gehören zu den klassichen Problemen der konvexen Geometrie (siehe z.B. Gritzmann und Klee zum Enthaltensein von Polytopen [25], Freund und Orlin zum Enthaltensein von Kugeln in Kugeln [16], oder Mangasarian zum Enthaltensein von konvexen Mengen in umgekehrt konvexen Mengen [62]).

Sei $\mathcal{S}_k$ die Menge der reellen, symmetrischen $k \times k$-Matrizen. Für Matrizen $A_0, \ldots, A_n \in \mathcal{S}_k$, bezeichnen wir mit $A(x)$ das lineare Matrixpolynom $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n \in \mathcal{S}_k[x]$. Die Menge

$$S_A := \{x \in \mathbb{R}^n \mid A(x) \succeq 0\}$$

nennt man Spektraeder, wobei $A(x) \succeq 0$ bedeutet, dass die Matrix $A(x)$ positiv semidefinit ist.

Wir untersuchen Enthaltenseinsfragen von Polyedern und Spektraedern. Da jedes Polyeder auch ein Spektraeder ist, stellt folgende Frage den Ausgangspunkt unserer Überlegungen dar: Gegeben zwei lineare Matrixpolynome $A(x) \in \mathcal{S}_k[x]$ und $B(x) \in \mathcal{S}_l[x]$, ist $S_A \subseteq S_B$?

Für Polytope ist die algorithmische Geometrie und Komplexität von Enthaltenseinsfragen bereits eingehend untersucht, hierbei sind insbesondere die Arbeiten von Gritzmann und Klee [23, 24, 25] hervorzuheben. Die algorithmische Komplexität von polytopalen Enthaltenseinsfragen hängt stark von der Art der Eingabedaten ab. So ist die Enthaltenseinsfrage für den Fall, dass beide Polytope über ihre Eckenmenge beschrieben werden ($\mathcal{V}$-Polytop), oder beide Polytope als Schnitt von Halbräumen beschrieben werden ($\mathcal{H}$-Polytop), in Polynomialzeit zu entscheiden. Andererseits ist die Frage des Enthaltenseins eines $\mathcal{H}$-Polytops in einem $\mathcal{V}$-Polytop co-NP-schwer (siehe [16, 25]).

Für Spektraeder ist weitaus weniger bekannt. Ben-Tal und Nemirovski untersuchten das „matrix cube problem“, das der Frage des Enthaltenseins eines Spektraeders in einem Würfel entspricht. Helton, Klep und McCullough [28] leiteten Ergebnisse zu Enthaltenseinsfragen von Spektraedern aus einer verallgemeinerten Fragestellung zu matrixwertigen Abbildungen ab.

Wir erweitern in dieser Arbeit die bestehenden Klassifikationen zur algorithmischen Komplexität von Polyedern auf den Fall von Polyedern und Spektraedern. Insbesondere kann die Frage, ob ein $\mathcal{V}$-Polytop in einem Spektraeder enthalten ist, in Polynomialzeit beantwortet werden. Die Frage nach dem Enthaltensein eines Spektraeders in einem $\mathcal{H}$-Polytop kann als semidefinites Lösbarkeitsproblem mit strikten Ungleichungen beschrieben werden. Die übrigen Fälle sind im Wesentlichen co-NP-schwer. Dies schließt die Frage nach dem Enthaltensein eines $\mathcal{H}$-Polytops in einem Spektraeder ein, bereits dann wenn das Spektraeder eine Kugel ist. Tabelle 1 fasst die entsprechenden Ergebnisse zusammen.

| | $\mathcal{H}$ | $\mathcal{V}$ | $\mathcal{S}$ |
|---|---|---|---|
| $\mathcal{H}$ | P | co-NP-vollständig | co-NP-schwer |
| $\mathcal{V}$ | P | P | P |
| $\mathcal{S}$ | „SDP“ | co-NP-schwer | co-NP-schwer |

Tabelle 1: Algorithmische Komplexität von Enthaltenseinsfragen. Die Zeilen bezeichnen die innere Menge, die Spalten die äußere Menge. $\mathcal{S}$ steht für Spektraeder.

Zum Beantworten der co-NP-schweren Fälle sind Relaxierungsmethoden von besonderem Interesse. Wir schlagen die Verwendung einer Hierarchie von hinreichenden semidefiniten Bedingungen für das Enthaltensein eines Spektraeders in einem zweiten vor. Dazu formulieren wir das Problem zunächst als polynomielles Optimierungsproblem über einem Spektraeder und wenden dann Positivstellensatzrelaxierungen für polynomielle Matrixungleichungen (basierend auf den Arbeiten von Kojima [43], Hol und Scherer [33] sowie Henrion and Lasserre [31]) an.

In diesem Zusammenhang zeigen wir das folgende Konvergenzresultat.

**Theorem 5.7.** *Die Optimalwerte der Hierarchie von Relaxierungen konvergieren gegen den Optimalwert des ursprünglichen polynomiellen Optimierungsproblems.*

Jede Stufe der Relaxierungshierarchie stellt eine hinreichende Bedingung für die Enthaltenseinsfrage dar. Insbesondere zeigen wir, dass jede dieser Relaxierungen mindestens so gut ist wie ein zuvor in [28, 40] eingeführtes Lösbarkeitskriterium.

**Theorem 5.10.** *Wenn das semidefinite Lösbarkeitsproblem aus [28, 40] eine Lösung hat und damit das Enthaltensein zweier Spektraeder zertifiziert, so zertifiziert auch die erste Relaxierungsstufe der Hierarchie das Enthaltensein.*

Dies erlaubt uns, die Exaktheitsaussagen aus [40] zu übertragen. Diese gelten insbesondere bereits für die kleinst mögliche Relaxierungsstufe unserer Hierarchie. Unter anderem zeigen wir, dass das Enthaltensein eines Spektraeders in einem Polyeder (in gegebener Normalform) durch jede Stufe der Relaxierung exakt charakterisiert wird.

Wir verdeutlichen die Effektivität der Methode anhand von numerischen Experimenten zu verschiedenen Enthaltenseinsproblemen. Dabei zeigt sich, dass das Verfahren sowohl auf einfachen Testproblemen, als auch auf zufällig generierten Problemen sehr gute Ergebnisse liefert. In vielen Beispielfällen wird das Enthaltensein bereits im ersten Relaxierungsschritt zertifiziert.

Das bislang beschriebene Verfahren liefert lediglich hinreichende Bedingungen für das Enthaltensein zweier Spektraeder. Wir schlagen daher ein auf den Branch and Bound-Verfahren aus dem ersten Teil basierendes Verfahren vor, welches im Falle des Nicht-Enthaltenseins der Spektraeder Punkte generiert, die als Zertifikat für das Nicht-Enthaltensein dienen.

# Acknowledgements

It is a pleasure for me to acknowledge the help I have received from my family, friends and colleagues during the preparation of this thesis.

My deepest thanks go to my advisor Thorsten Theobald. Even in busy times he always had an open ear and took the time to discuss ideas and problems with me. With his broad knowledge and good intuition, he pointed me in the right direction when necessary.

I was fortunate to have Kai Kellner as my colleague. We often complemented each other, sometimes struggled together, but in the end our collaboration on containment problems was always very fruitful.

My colleagues from the institute of mathematics in Frankfurt helped me with discussions, comments and proofreading for this thesis, but also offered distraction and friendship. Thank you, Hartwig Bosse, Mareike Dressler, Christian Haase, Jan Hoffmann, Sadik Iliman, Thorsten Jörgens, Martina Juhnke-Kubitzke, Kai Kellner, Matthias Riedel, Björn Ulbricht, and Timo de Wolff, you made it a pleasure to work here.

I wish to thank my parents Sonja and Wilhelm Trabandt for their trust and support throughout the time of my studies.

Corinna Koch helped me to get back on track, when motivation was low and gave me her full support during all phases of this work. Thank you for believing in me.

Finally, financial support from the "Studienstiftung des Deutschen Volkes" is greatly acknowledged.

# Contents

# 1 Introduction

This work is concerned with two topics at the intersection of convex algebraic geometry and optimization.

We develop a new method for the optimization of polynomials over polytopes. From the point of view of convex algebraic geometry the most common method for the approximation of polynomial optimization problems is to solve semidefinite programming relaxations coming from the application of Positivstellensätze. In optimization, non-linear programming problems are often solved using branch and bound methods. We propose a fused method that uses Positivstellensatz-relaxations as lower bounding methods in a branch and bound scheme. By deriving a new error bound for Handelman's Positivstellensatz, we show convergence of the resulting branch and bound method.

Through the application of Positivstellensätze, semidefinite programming has gained importance in polynomial optimization in recent years. While it arises to be a powerful tool, the underlying geometry of the feasibility regions (spectrahedra) is not yet well understood. See [8] for an overview on semidefinite programming, the underlying geometry and applications to polynomial optimization. In this work, we study polyhedral and spectrahedral containment problems, in particular we classify their complexity and introduce sufficient criteria to certify the containment of one spectrahedron in another one.

### Positivstellensatz Branch and Bound Methods

We study polynomial optimization problems over a semialgebraic set (sets defined by polynomial inequalities). The goal is to find the minimum of a polynomial objective function $p(x) \in \mathbb{R}[x_1, \ldots, x_n]$ over a semialgebraic set $K$, i.e.,

$$\begin{aligned} \text{minimize} \quad & p(x) \\ \text{s.t.} \quad & x \in K. \end{aligned} \tag{1.1}$$

Such polynomial optimization problems occur in many applications such as finance, structural engineering and control. Moreover, many important combinatorial problems can be modelled as a polynomial optimization problem. These problems are, in general, very hard to solve.

Following the pioneering work of Lasserre [46] and Parrilo [70, 71], it has become common practice in convex algebraic geometry to rewrite problem (1.1) in its dual

form

$$\begin{aligned} \text{maximize} \quad & t \\ \text{s.t.} \quad & p(x) - t \geq 0, \\ & x \in K, \end{aligned}$$

and to replace the constraints by weaker, but more tractable conditions using Positivstellensätze, see [51, 53]. By bounding the degree of the Positivstellensatz certificates, one attains a hierarchy of relaxations, which can be solved using linear or semidefinite programming methods, depending on the Positivstellensatz.

We mainly focus on the case where $K$ is a polytope and Handelman's Positivstellensatz can be applied. It states that a polynomial $p(x) \in \mathbb{R}[x]$ is positive on a polytope $P = \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_k(x) \geq 0, \text{ all linear}\}$ if and only if $p(x)$ has a representation of the form

$$p(x) = \sum_{\alpha \in \mathbb{N}^n} \lambda_\alpha g_1^{\alpha_1}(x) \cdots g_k^{\alpha_k}(x), \quad \lambda_\alpha \geq 0. \tag{1.2}$$

Restricting the degree of the right hand side in equation (1.2) yields a linear programming problems, which can be solved very efficiently.

The drawback of this approach is that the complexity of the relaxation increases drastically when the order of the relaxation is increased by choosing a higher degree bound for the Positivstellensatz. In this work we discuss a method that uses Positivstellensatz relaxations, but instead of increasing the order of the relaxation to obtain more precise results, we propose to divide the feasibility set $K$. The resulting subproblems are solved at a fixed relaxation order, keeping the complexity of the problems constant, but increasing the number of problems that have to be solved.

Our main contribution in this area is the generalization of an error bound for relaxations coming from Handelman's Positivstellensatz by De Klerk and Laurent [13]. We extend their result for the optimization of polynomials over the unit cube to arbitrary hyperrectangular sets and, in a restricted setting, to general polytopes, making use of earlier results by Garloff [18] on Bernstein approximations. The bound depends on the length of the longest edge of a bounding box of the polytope. This property is later used to derive theoretical results on the convergence of the branch and bound scheme.

The elementary branch and bound method studied in this work uses a relaxation based on Handelman's Positivstellensatz of the lowest possible order as lower bounding method. The upper bound is computed by evaluating the objective on a feasible point. Both bounds converge to the optimal value as the sets under consideration are successively bisected along the longest edge. Since lower and upper bounds are computed, it results in an approximation of the optimum of certified accuracy.

To show the effectiveness of the approach, we apply the branch and bound scheme to a number of sample problems. Our numerical experiments suggest that

the branching method does indeed yield significantly improved results, already when adding a single cut, compared to the unbranched problem at the same relaxation order.

### Containment Problems for Spectrahedra

The second topic studied in this thesis is about containment problems of spectrahedra. Containment problems of convex sets belong to the classical problems in convex geometry (see, e.g., Gritzmann and Klee for the containment of polytopes [25], Freund and Orlin for containment problems of balls in balls [16], or Mangasarian for containment of convex sets in reverse-convex sets [62]).

Denote by $\mathcal{S}_k$ the set of all real symmetric $k \times k$-matrices. For $A_0, \ldots, A_n \in \mathcal{S}_k$, let $A(x)$ denote the *linear (matrix) pencil* $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n \in \mathcal{S}_k[x]$. Then the set

$$S_A := \{x \in \mathbb{R}^n \mid A(x) \succeq 0\}$$

is called a *spectrahedron*, where $A(x) \succeq 0$ denotes positive semidefiniteness of the matrix $A(x)$.

We study containment problems for polyhedra and spectrahedra. Since polyhedra are special cases of spectrahedra, we can use the following general setup: Given two linear pencils $A(x) \in \mathcal{S}_k[x]$ and $B(x) \in \mathcal{S}_l[x]$, is $S_A \subseteq S_B$?

For polytopes (i.e., bounded polyhedra), the computational geometry and computational complexity of containment problems have been studied in detail. See in particular the classifications by Gritzmann and Klee [23, 24, 25]. Notably, it is well known that the computational complexity of deciding containment problems strongly depends on the type of the input. For instance, if both polytopes are given by their vertices ($\mathcal{V}$-polytopes), or both polytopes are given as an intersection of halfspaces ($\mathcal{H}$-polytopes), containment can be decided in polynomial time, while it is co-NP-hard to decide whether an $\mathcal{H}$-polytope is contained in a $\mathcal{V}$-polytope (see [16, 25]).

For spectrahedra, significantly less is known. Ben-Tal and Nemirovski studied the matrix cube problem [6], which corresponds to the containment problem where $S_A$ is a cube. In a more general setting, Helton, Klep, and McCullough [28] studied containment problems of matricial positivity domains (which live in a union of spaces of different dimensions). As a byproduct, they also derive some implications for containment of spectrahedra.

In this work, we extend existing complexity classifications for the polyhedral situation to the situation where polytopes and spectrahedra are involved. In particular, the containment question of a $\mathcal{V}$-polytope in a spectrahedron can be decided in polynomial time, and the question whether a spectrahedron is contained in an $\mathcal{H}$-polytope can be formulated by the complement of semidefinite feasibility problems (involving also strict inequalities). Roughly speaking, all other cases are co-NP-hard. This includes the containment problem of an $\mathcal{H}$-polytope in a spectrahedron, already when the spectrahedron is a ball.

To deal with the situation that the general containment problem for spectrahedra is co-NP-hard, relaxation techniques are of particular interest. We propose and study a hierarchy of sufficient semidefinite conditions to certify the containment of a spectrahedron in another one. This approach comes from applying a moment relaxation to a suitable polynomial optimization formulation. We use common relaxation techniques (by Kojima [43], Hol and Scherer [33] as well as Henrion and Lasserre [31]) to derive a (sufficient) semidefinite hierarchy for the containment problem. The semidefinite hierarchy provides a much more comprehensive approach towards the containment problems than the earlier sufficient criterion studied by Helton, Klep and McCullough [28] and by Kellner, Theobald and Trabandt [40].

Our main contributions in this area are the following.

1. Based on polynomial matrix inequalities, we provide a hierarchy of sufficient semidefinite criteria for the containment problem and prove that the sequence of optimal values converges to the optimal value of the underlying polynomial optimization problem.

2. Any relaxation step of the hierarchy yields a sufficient criterion for the containment problem. We prove that each of these sufficient criteria is at least as powerful as an earlier criterion introduced in [28, 40], in the sense that whenever the criterion of [28, 40] is satisfied, then also the criterion from any of the relaxation steps of the hierarchy is satisfied. In particular, this already holds for the criterion coming from the initial relaxation step. This allows to carry all exactness results from [40] forward to our new hierarchical approach, in particular it shows that the sufficient criterion for the containment of spectrahedra in polyhedra (in normal form) is an *exact* characterization.

3. We demonstrate the effectiveness of the approach by providing numerical results for several containment problems and radii computations.

4. Bearing in mind the insights from the first part, we apply an adapted branching method to some instances of the containment problem, allowing us to find certificates for non-containment.

## Thesis overview

This work is structured as follows.

Chapter 2 introduces notation and the necessary background on optimization and Positivstellensatz relaxations. Following the geometrical characater of this work we introduce polyhedra and spectrahedra before discussing the fundamentals of linear and semidefinite programming. We review several Positivstellensätze and resulting relaxations for polynomial optimization, discussing methods for polynomial matrix inequalities in a separate section.

In Chapter 3, we first review error bounds for Positivstellensatz relaxations, before focussing on the new results concerning Handelman's Positivstellensatz. For clarity, the error bound for the Handelman approach is derived first in the univariate setting and then extended to the multivariate setting. We make use of

Bernstein approximations to derive the error bound and also discuss similarities of the Handelman and the Bernstein approach.

Our branch and bound scheme based on Handelman's Positivstellensatz relaxations are discussed in Chapter 4. We begin more generally by studying the impact of bisecting the feasibility set of a polynomial optimization problem before proposing our branch and bound method. This is also where we discuss general theory on the convergence of branch and bound schemes and show convergence of the proposed approach.

Chapter 5 is devoted to spectrahedral containment problems. After classifying their complexity, our relaxation hierarchy is introduced. We relate it to the earlier criterion introduced in [28, 40] and apply it to a number of test cases. The final section of this chapter connects the two topics of this thesis. We explain how a modified version of the branch and bound method can be used to derive certificates of non-containment for given spectrahedra.

We conclude this thesis in Chapter 6 with a discussion of open problems.

## Numerical Computations

All numerical computations described in this thesis were performed on a desktop computer with Intel Core i3-2100 @ 3.10 GHz and 4 GB of RAM.

For our computations, we implemented the branch and bound methods as well as the containment hierarchy using high-level YALMIP [58, 59] code. We used MOSEK 7 [3] as an external solver for the optimization problems defined in YALMIP. The MATLAB version used was R2011b.

# 2 Notation and Preliminaries on Mathematical Optimization

In this section, we introduce the basic concept of mathematical optimization. We use common notation, as it can be found, e.g., in [9]. In the following sections, we discuss special instances, such as linear, semidefinite and polynomial optimization.

We consider *mathematical optimization problems* of the form

$$\begin{aligned} \text{minimize} \quad & p(x) \\ \text{s.t.} \quad & f_i(x) \geq 0, \quad i \in \{1, \ldots, k\}. \end{aligned} \tag{2.1}$$

Here, $x$ is the vector of *optimization variables*, the function $p : \mathbb{R}^n \to \mathbb{R}$ is the *objective* function and the functions $f_i : \mathbb{R}^n \to \mathbb{R},\ i \in [m]$ are the *constraints*.

The set $\{x \in \mathbb{R}^n \mid f_i(x) \geq 0,\ \forall i \in \{1, \ldots, k\}\}$ is called the *feasible set* or *feasibility region* of the optimization problem. A vector in this set is called a *feasible solution*. The value $p(x)$ is called the *objective value* of the vector $x$. A vector $x^*$ that has the smallest objective value among all feasible solutions is called *optimal*, the objective value $p(x^*)$ of $x^*$ is the optimal value of the optimization problem (2.1). The goal is to *solve* the optimization problem, that means to find its optimal value and sometimes also an optimal vector.

The optimization problem stated in its general form (2.1) is in general very hard to solve. For special classes of the problem, however, efficient algorithms are known. Linear and semidefinite programming problems are such special cases. We will discuss them in section 2.2. For polynomial optimization problems, which we will introduce in section 2.3, on the other hand no efficient exact algorithms are known. Thus we also discuss relaxation methods for these problems, which offer a trade-off between exactness and complexity of the algorithms.

In this work, we approach most problems from a geometric point of view. In the same spirit, we talk about linear and semidefinite programming problems in this chapter by introducing the respective feasibility sets at first. Before diving into mathematical programming, we consolidate the notation that we use throughout in the next section.

## 2.1 Notation

The symbols $\mathbb{N}, \mathbb{Q}$, and $\mathbb{R}$ denote the non-negative integers, the field of rational numbers and the field of real numbers. $\mathbb{R}_+$ stands for the subset of non-negative numbers in $\mathbb{R}$. We set $\mathbb{N}^n_t := \{\alpha \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \leq t\}$ for $t \in \mathbb{N}$.

The set of real $n$-vectors is denoted by $\mathbb{R}^n$, the set of $m \times n$ matrices by $\mathbb{R}^{m\times n}$. $\|\cdot\|$ denotes the standard Euclidean norm. By $\mathbb{B}_r(p)$, we denote the (closed) Euclidean ball with center $p$ and radius $r > 0$.

For a vector $x \in \mathbb{R}^n$, we refer to the $i$-th entry of $x$ by $x_i$. For vectors $a, b \in \mathbb{R}^n$, inequalities are meant to be understood component-wise, for example $a \geq b$ stands for $a_1 \geq b_1, \ldots, a_n \geq b_n$.

For a matrix $A$, the $(i, j)$-th entry of $A$ is labeled by $a_{ij}$. For a block matrix $B$, we label the $(i, j)$-th block by $B_{ij}$ and the $(s, t)$-th entry of $B_{ij}$ by $(B_{ij})_{st}$.

A square matrix with 1 in the entry $(i, j)$ and zeros otherwise is denoted by $E_{ij}$. The $n \times n$ identity matrix is denoted by $I_n$.

The symmetric $k \times k$-matrices are denoted by $\mathcal{S}_k$, the subset of positive semidefinite matrices by $\mathcal{S}_k^+ = \{X \in \mathcal{S}_k \mid X \succeq 0\}$, where for two matrices $A, B \in \mathcal{S}_k$, the notation $A \succeq B$ is a shorthand for $A - B$ being positive semidefinite. The space $\mathcal{S}_k$ is equipped with the trace inner product $\langle A, B\rangle := \mathrm{Tr}(AB^T) = \sum_{i,j} a_{ij}b_{ij}$, that can also be interpreted as the usual Euclidean inner product on $\mathbb{R}^{k\times k}$.

Recall that a matrix $A$ is positive semidefinite, if and only if one of the following equivalent statements holds.

- $z^T Az \geq 0 \quad \forall z \in \mathbb{R}^k$;
- All eigenvalues of $A$ are non-negative;
- All principal minors of $A$ are non-negative.

The *Kronecker product* $A \otimes B$ of square matrices $A$ of size $k \times k$ and $B$ of size $l \times l$ is the $kl \times kl$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}\,B & \ldots & a_{1k}\,B \\ \vdots & \ddots & \vdots \\ a_{k1}\,B & \ldots & a_{kk}\,B \end{bmatrix} \tag{2.2}$$

(see, e.g., [12, 34]). It is well known (see, e.g., [34, Cor. 4.2.13]) that the Kronecker product of two positive semidefinite matrices is again positive semidefinite.

The ring of multivariate polynomials in $n$ variables is denoted by $\mathbb{R}[x_1, \ldots, x_n]$ or by $\mathbb{R}[x]$ for short. For $\alpha \in \mathbb{N}^n$, we abbreviate the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ by $x^\alpha$. The degree of the monomial $x^\alpha$ is given by $|\alpha| = \sum_{i=1}^n \alpha_i$.

A polynomial is a finite linear combination of monomials $p(x) = \sum_{\alpha\in\mathbb{N}^n} p_\alpha x^\alpha \in \mathbb{R}[x]$, where only finitely many $p_\alpha$ are nonzero, the degree of $p$ is the maximum degree over all appearing monomials, $\deg(p) := \max(|\alpha| \mid p_\alpha \neq 0)$. The set of all polynomials of degree less than or equal to $t$ is denoted by $\mathbb{R}[x]_t = \{p(x) \mid p(x) = \sum_{\alpha\in\mathbb{N}^n_t} p_\alpha x^\alpha\}$, in the same spirit, we define $\mathbb{R}[x]_{m,\ldots,m}$ to be the multivariate polynomials of degree at most $m$ in each variable. A polynomial in which all terms are of the same degree is called a *homogeneous polynomial*. We define the vector of coefficients of $p$ as $\vec{p} = (p_\alpha)_{\alpha\in\mathbb{N}^n}$.

We will use the notation for monomials also for sets of polynomials. Given $g_1, \ldots, g_k \in \mathbb{R}[x]$ and $\alpha \in \mathbb{N}^k$, we define $g^\alpha := \prod_{j=1}^k g_j^{\alpha_j}$.

Let $\mathcal{S}_k[x]$ be the set of symmetric $k \times k$-matrices with polynomial entries in $x = (x_1, \ldots, x_n)$. For $A_0, \ldots, A_n \in \mathcal{S}_k$, denote by $A(x)$ the *linear (matrix) pencil*

$A(x) = A_0 + x_1A_1 + \cdots + x_nA_n \in \mathcal{S}_k[x]$. We call the generalized inequality $A(x) \succeq 0$ a *linear matrix inequality.*

A subset $C$ of a vector space $V$ is a convex cone if $\alpha x + \beta y$ belongs to $C$, for any non-negative scalars $\alpha,\ \beta$, and any $x,\ y$ in $C$. The following sets are convex cones.

- The vectors with non-negative entries in $\mathbb{R}^n$;
- $\mathcal{S}_k \subset \mathbb{R}^{k\times k}$;
- The set of non-negative polynomials in $n$ variables $\mathcal{P}_n \subset \mathbb{R}[x_1 \dots x_n|$;
- The set of sums of squares polynomials in $n$ variables $\Sigma_n \subset \mathbb{R}[x_1 \dots x_n|$, as defined in section 2.3.1.

We define the index sets $[d] := \{1, \dots, d\}$ and $[d]_0 := \{0, \dots, d\}$, as well as the corresponding n-tuples $[d]^n := \{(d_1, \dots, d_n) \mid d_1 \in [d], \dots, d_n \in [d]$ and $[d]_0^n$ analogously. When comparing $n$-tuples, inequalities are again meant to be understood component-wise.

## 2.2 Linear and Semidefinite Programming

In this section, we will encounter special classes of optimization problems. Linear programs are problems of the form (2.1), where both the objective and the constraint functions are affine linear functions. We discuss these problems in section 2.2.1. Semidefinite programs are problems where we optimize a linear objective function over a set defined by matrix inequalities, as described in section 2.2.2.

### 2.2.1 Polyhedra, Polytopes and Linear Programming

A set $P \subset \mathbb{R}^n$ is called a *polyhedron* if it is the intersection of finitely many affine halfspaces. Algebraically, these are described by linear inequalities. A polyhedron $P$ is given by a matrix $A \in \mathbb{R}^{k\times n}$ and a vector $b \in \mathbb{R}^n$ such that

$$P = \{x \in \mathbb{R}^n \mid Ax + b \geq 0\}. \tag{2.3}$$

A bounded polyhedron is called *polytope.*

If an inequality $c^Tx + d \geq 0$ is implied by the inequalities $Ax + b \geq 0$, or, more precisely, if $c^Tx + d \geq 0$ holds for each $x$ satisfying $Ax + b \geq 0$, we call it redundant in the representation of the polytope $P$. Farkas' Lemma characterizes the redundant inequalities.

**Proposition 2.1** (Affine form of Farkas' Lemma, [83, Corollary 7.1h])**.** *Let the polyhedron $P = \{x \in \mathbb{R}^n : Ax + b \geq 0\}$ be nonempty and $c^Tx + d \geq 0$ be a redundant inequality. Then for some $d' \leq d$, the linear inequality $c^Tx + d' \geq 0$ is a non-negative linear combination of the inequalities in the system $Ax + b \geq 0$.*

A *linear programming problem* is the problem of minimizing a linear objective function over a polyhedron. We use the following standard form for linear programs.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & Ax + b \geq 0. \end{aligned} \tag{2.4}$$

We call the linear program (2.4) the *primal* linear program. Associated with this linear program is the *dual* linear program

$$\begin{aligned} \text{maximize} \quad & -y^T b \\ \text{s.t.} \quad & A^T y = c \\ & y \geq 0 \\ & y \in \mathbb{R}^k. \end{aligned} \tag{2.5}$$

The two linear programs are linked by the following duality theory. The key property of the dual problem is that it bounds the objective value of the primal problem in the following sense.

**Theorem 2.2** (Weak duality [38, Theorem 4.11])**.** *Let $x$ be a feasible solution for the primal linear program and $y$ a feasible solution for the dual linear program. Then $c^T x \geq -b^T y$.*

**Theorem 2.3** (Strong duality [38, Theorem 4.13])**.** *Given a pair of primal-dual optimization problems*

$$\min\{c^T x \mid Ax + b \geq 0\} \quad \textit{and} \quad \max\{-y^T b \mid A^T y = c,\ y \geq 0\},$$

*exactly one of the following statements is true.*

- *Both problems are feasible and the optimal values coincide.*
- *One of the problems is infeasible, the other unbounded.*
- *Both problems are infeasible.*

Infeasibility of a linear program is characterized by another version of Farkas' Lemma.

**Theorem 2.4** ([83, Corollary 7.1e])**.** *Either the polytope $P = \{x \in \mathbb{R}^n \mid Ax+b \geq 0\}$ is non-empty or the following system has a solution $y \in R^k$*

$$y \geq 0, \quad y^T A = 0, \quad y^T b < 0. \tag{2.6}$$

To decide if $P$ is empty, we can solve the following linear program

$$\begin{aligned} \text{minimize} \quad & y^T b \\ \text{s.t.} \quad & y^T A = 0 \\ & y \geq 0. \end{aligned} \tag{2.7}$$

$P$ is empty if and only if the optimal value of this problem is non-negative.

An alternative way of determining whether $P$ is empty is to solve the following auxiliary problem

$$\begin{aligned} \text{maximize} \quad & y \\ \text{s.t.} \quad & Ax + b \geq y \\ & y \leq 0. \end{aligned} \tag{2.8}$$

To decide feasibility, note that $P$ is nonempty if and only if the optimal value of (2.8) is zero. If it is zero, the projection of any optimal point to the $x$-coordinates lies in $P$. A similar approach is often taken when solving linear programs with the simplex method to find an initial feasible vertex.

A special class of polytopes that we will use extensively are $n$-dimensional boxes. For $a, b \in \mathbb{R}^n, a \leq b$, define

$$\begin{aligned} [\underline{a}, \underline{b}] &:= [a_1, b_1] \times \ldots \times [a_n, b_n] \\ &= \{x \in \mathbb{R}^n \mid g_{l_i}(x_i) = x_i - a_i \geq 0,\ g_{u_i}(x) = b_i - x_i \geq 0,\ i = 1, \ldots, n\}. \end{aligned} \tag{2.9}$$

While the optimization problem (2.1) is, in general, hard to solve, the linear version (2.4) can efficiently be solved and is used in many practical applications.

**Theorem 2.5** (Khachiyan's theorem, [83, Theorem 13.4])**.** *Linear programming problems with rational data can be solved in polynomial time.*

Until now, we described polyhedra as the intersection of finitely many halfspaces. A bounded polyhedron can also be described as the convex hull of finitely many points in $\mathbb{R}^n$. For algorithmic questions in $n$-dimensional space it is crucial whether a polytope is given in the first way ($\mathcal{H}$*-polytope*) or in the second way ($\mathcal{V}$*-polytope*). Our model of computation is the binary Turing machine: polytopes are presented by certain rational numbers, and the size of the input is defined as the length of the binary encoding of the input data (see, e.g., [23]). A $\mathcal{V}$-polytope $P$ is given by a tuple $(n; m; v^{(1)}, \ldots, v^{(m)})$ with $n, m \in \mathbb{N}$, and $v^{(1)}, \ldots, v^{(m)} \in \mathbb{Q}^n$ such that $P = \operatorname{conv}\{v^{(1)}, \ldots, v^{(m)}\}$. An $\mathcal{H}$-polytope $P$ is given by a tuple $(n; k; A; b)$ with $n, k \in \mathbb{N}$, a rational $k \times n$-matrix $A$, and $b \in \mathbb{Q}^k$ such that $P = \{x \in \mathbb{R}^n \mid b + Ax \geq 0\}$ is bounded. If the $i$-th row $(b + Ax)_i \geq 0$ defines a facet of $P$, then the $i$-th row of $A$ is an *inner* normal vector of this facet.

For fixed dimension, $\mathcal{H}$- and $\mathcal{V}$-representations of a rational polytope can be converted into each other in polynomial time. In general dimension (i.e., if the dimension is not fixed but part of the input) the size of one representation can be exponential in the size of the other [63].

This section on linear programming is loosely based on the book by Joswig and Theobald [38], which is an excellent resource for the geometry of polytopes and spectrahedra and the book by Schrijver [83], which gives a comprehensive introduction into the theory of linear programming.

### 2.2.2 Spectrahedra and Semidefinite Programming

Given matrices $A_0, A_1, \dots A_n \in \mathcal{S}_k$, and the *linear (matrix) pencil* $A(x) = A_0 + \sum_{p=1}^{n} x_p A_p \in \mathcal{S}_k[x]$ we define a *spectrahedron* $S_A$ as the positivity region of the linear matrix pencil $S_A = \{x \in \mathbb{R}^n \mid A(x) \succeq 0\}$.

The equivalence between positive definiteness of $A_0$ and the origin being an interior point is not true. Moreover, in general, the interior of $S_A$ does not coincide with the positive definiteness region of the pencil. However, if the spectrahedron $S_A$ has nonempty interior (or, equivalently, $S_A$ is full-dimensional), then there exists a *reduced* linear pencil that is positive definite exactly on the interior of $S_A$.

**Proposition 2.6** ([20, Corollary 5])**.** *Let $S_A = \{x \in \mathbb{R}^n \mid A(x) \succeq 0\}$ be full-dimensional and let $N$ be the intersection of the nullspaces of $A_i$, $i = 0, \dots, n$. If $V$ is a basis of the orthogonal complement of $N$, then $S_A = \{x \in \mathbb{R}^n \mid V^T A(x) V \succeq 0\}$ and the interior of $S_A$ is $\operatorname{int}(S_A) = \{x \in \mathbb{R}^n \mid V^T A(x) V \succ 0\}$.*

Furthermore, the spectrahedron $S_A$ contains the origin in its interior if and only if there is a linear pencil $A'(x)$ with the same positivity domain such that $A'_0 = I_k$; see [29]. To simplify notation, we sometimes assume that $A(x)$ is of this form and refer to it as a *monic* linear pencil, i.e., $A_0 = I_k$.

Given the linear pencils $A(x) \in \mathcal{S}_k[x]$, we call the linear pencil

$$\widehat{A} = 1 \oplus A(x) = 1 \oplus A_0 + \sum_{p=1}^{n} x_p (0 \oplus A_p) \tag{2.10}$$

the *extended linear pencil* of $A(x)$, where $\oplus$ denotes the direct sum of matrices.

Note that every polyhedron $P = \{x \in \mathbb{R}^n \,|\, b + Ax \geq 0\}$ has a natural representation as a spectrahedron:

$$P = P_A = \left\{ x \in \mathbb{R}^n \;:\; A(x) = \begin{bmatrix} a_1(x) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & a_k(x) \end{bmatrix} \succeq 0 \right\}, \tag{2.11}$$

where $a_i(x)$ abbreviates the $i$-th entry of the vector $b + Ax$. $P_A$ contains the origin if and only if the inequalities can be scaled so that $b = \mathbb{1}_k$, where $\mathbb{1}_k$ denotes the all-ones vector in $\mathbb{R}^k$. Hence, in this case, $A(x)$ is monic, and it is called the *normal form* of the polyhedron $P_A$.

A centrally-symmetric ellipsoid with axis-aligned semi-axes of lengths $a_1, \dots, a_n$ can be written as the spectrahedron $S_A$ of the monic linear pencil

$$A(x) \;=\; I_{n+1} + \sum_{p=1}^{n} \frac{x_p}{a_p} (E_{p,n+1} + E_{n+1,p}). \tag{2.12}$$

We call (2.12) the *normal form of the ellipsoid*. Specifically, for the case of equal semi-axis lengths $r := a_1 = \cdots = a_n$ this gives the *normal form of a ball* with radius $r$.

Similarly to linear programs, we define a *semidefinite programming problem* as the problem of minimizing a linear objective function over a spectrahedron. We use the following standard form for semidefinite programs.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{s.t.} \quad & A(x) \succeq 0. \end{aligned} \tag{2.13}$$

Again there is a dual problem associated to the primal formulation above.

$$\begin{aligned} \text{maximize} \quad & -\langle A_0, Y \rangle \\ \text{s.t.} \quad & \langle A_i, Y \rangle = c_i, \ i = 1, \ldots, n, \\ & Y \succeq 0. \end{aligned} \tag{2.14}$$

Here the variable is the matrix $Y$, which is subject to $m$ equality constraints and the matrix non-negativity condition. While it is not completely obvious, the dual problem is also a semidefinite program, it can be transformed to the form of the primal problem (2.13).

It is worthwhile to compare the formulations of the dual linear program (2.5) to the dual semidefinite program (2.14). In both cases, we have a linear objective function and linear equality constraints. The only difference is the non-negativity constraint. In the linear program, the variables are constrained to the non-negative orthant, that is the cone of non-negative vectors, in the semidefinite program to the cone of positive semidefinite matrices. With this regard, both problems can be seen as instances of a conic optimization problem.

As in linear programming, there exists a rich duality theory for semidefinite programs.

**Theorem 2.7** (Weak duality [88, p. 64])**.** *Let $x$ be a feasible solution for the primal semidefinite program and $Y$ a feasible solution for the dual semidefinite program. Then $c^T x \geq -\langle A_0, Y \rangle$.*

While weak duality translates directly to the new setting, strong duality requires some more prerequisites.

We denote the optimal values of the primal and dual optimization problems by $p^* := \inf\{c^T x \mid A(x) \succeq 0\}$, and $d^* := \sup\{-\langle A_0, Y \rangle \mid \langle A_i, Y \rangle = c_i, \ i = 1, \ldots, n, \ Y \succeq 0\}$. Furthermore, let $X_{opt}$ and $Y_{opt}$ denote the optimal sets, $X_{opt} := \{x \mid A(x) \succeq 0, \ c^T x = p^*\}$ and $Y_{opt} := \{Y \mid -\langle A_0, Y \rangle \mid \langle A_i, Y \rangle = c_i, \ i = 1, \ldots, n, \ Y \succeq 0, \ -\langle A_0, Y \rangle = d^*\}$.

**Theorem 2.8** (Strong duality [88, Theorem 3.1.])**.** *Let $p^*$ be the optimal value of a semidefinite program and $d^*$ be the optimal value of the corresponding dual semidefinite program. Then $p^* = d^*$ if either of the following conditions hold.*

- *The primal problem is strictly feasible, i.e., there exists an $x$ such that $A(x)$ is positive definite;*
- *The dual problem is strictly feasible, i.e., there exists $Y$ with $Y = Y^T \succ 0$, $\langle A_i, Y \rangle = c_i, \ i = 1, \ldots, n$.*

*If both conditions hold, the optimal sets $X_{opt}$ and $Y_{opt}$ are nonempty.*

For algorithmic questions, a linear pencil is given by a tuple $(n; k; A_0, \ldots, A_n)$ with $n, k \in \mathbb{N}$ and $A_0, \ldots, A_n$ rational symmetric matrices.

A Semidefinite Feasibility Problem (SDFP) is defined as the following decision problem (see, e.g., [76]): Given a linear pencil defined by a tuple $(n; k; A_0, \ldots, A_n)$ with $n, k \in \mathbb{N}$ and $A_0, \ldots, A_n$ rational symmetric matrices. Are there real numbers $x_1, \ldots, x_n$ such that $A(x) = A_0 + \sum_{p=1}^{n} x_p A_p \succeq 0$, or equivalently, is the spectrahedron $S_A$ non-empty? Although semidefinite programs can be approximated up to an additive error of $\varepsilon$ in polynomial time, the question "SDFP $\in$ P?" is one of the major open complexity questions in semidefinite programming (see [12, 76]). In practice, however, SDFPs can be solved efficiently by semidefinite programming.

The material in this section is mostly based on the early article by Vandenberghe and Boyd [88]. For a thorough treatment we recommend the book by De Klerk [12].

## 2.3 Polynomial Optimization

Many practical problems can be formulated as polynomial optimization problems. But in most cases, these problems are non-convex and hard to handle. The recent approach to tackle these problems is to use sums of squares reformulations of the problems to solve or at least approximate the optimal value by linear or semidefinite programs. We will now introduce the main ideas of this approach.

Most material from this section can be found in greater detail in the excellent survey [53] by Monique Laurent or in Lasserre's book [51] and the Handbook on Semidefinite, Conic and Polynomial Optimization [4].

### 2.3.1 Unconstrained Polynomial Optimization and Sums of Squares

Starting with an unconstrained polynomial optimization problem of the form

$$\text{minimize} \quad p(x), \tag{2.15}$$

the first step to approach the problem is usually to reformulate it as a convex optimization problem. This can be achieved by introducing an additional variable and rewriting the problem as

$$\begin{aligned} \text{maximize} \quad & t, \\ \text{s.t.} \quad & p(x) - t \geq 0. \end{aligned} \tag{2.16}$$

The resulting problem is indeed convex, since the non-negative polynomials in $n$ variables form a convex cone. We denote this cone by $\mathcal{P}_n$ and the subset of non-negative polynomials of degree at most $d$ by $\mathcal{P}_{n,d} := \mathcal{P}_n \cap \mathbb{R}[x]_d$. Note that both problems have the same optimal value, which we denote by $p_{min}$.

The non-negativity condition in problem (2.16) is then replaced by a more tractable condition using sum of squares polynomials.

**Definition 2.9.** A polynomial $p(x) \in \mathbb{R}[x]$ is a *sum of squares* (SOS) if it can be written as

$$p(x) = \sum_{j=1}^{m} h_j(x), \quad h_j(x) \in \mathbb{R}[x].$$

The sums of squares in $n$ variables form a convex cone, which we denote by $\Sigma_n$, the intersection with the polynomials of degree at most $d$ is denoted by $\Sigma_{n,d} = \Sigma_n \cap \mathbb{R}[x]_d$. Remarkably, it can be verified whether a given polynomial is a sum of squares using semidefinite programming.

**Lemma 2.10.** *A polynomial $p(x) \in \mathbb{R}[x]_{2d}$ is a sum of squares if and only if the following system in the matrix variable $X = (X_{\alpha,\beta})_{\alpha,\beta\in\mathbb{N}^n_d}$ is feasible.*

$$\begin{aligned} X &\succeq 0 \\ \sum_{\substack{\alpha,\beta\in\mathbb{N}^n_d,\\ \alpha+\beta=\gamma}} X_{\alpha,\beta} &= p_\gamma, \quad |\gamma| \leq 2d. \end{aligned}$$

By requiring the polynomial $p(x) - t$ in the optimization problem to be a sum of squares of restricted degree instead of requiring it to be non-negative, we arrive at the following relaxed problem, which is a semidefinite program and can be solved efficiently.

$$\begin{aligned} \text{maximize} \quad & t, \\ \text{s.t.} \quad & p(x) - t \in \Sigma_{n,d}. \end{aligned} \tag{2.17}$$

The optimal value of this relaxation is denoted by $p^{(d)}_{sos}$. While every sum of squares is obviously non-negative, that is $\mathcal{P}_n \subset \Sigma_n$, and $\mathcal{P}_{n,d} \subset \Sigma_{n,d}$, it is a well-known result by Hilbert, that the reverse is not true in general.

**Theorem 2.11** ([32])**.** *$\mathcal{P}_{n,d} = \Sigma_{n,d}$ only in the following cases:*

- $n = 1$,
- $d = 2$,
- $(n, d) = (2, 4)$.

This means that problem (2.17) is indeed a relaxation to problem (2.16) and the inequality $p^{(d)}_{sos} \leq p_{min}$ may be strict.

On the other hand, Hilbert conjectured, that every non-negative polynomial can be written as a sum of squares of rational functions. This conjecture, which has become known as Hilbert's 17th problem, was proved to be correct by Emil Artin in 1926. This result can be used for our purposes. Clearing denominators we can look for a representation

$$gp = h$$

for some polynomials $g, h \in \Sigma$. While there exist bounds on the degree of $g$ and $h$, they are too large for practical implementations. But it is still a good idea to strengthen the relaxation (2.17) by multiplying the polynomial $p(x) - t$ with a polynomial known to be non-negative.

### 2.3.2 Semialgebraic Sets and Constrained Polynomial Optimization Problems

Let us now consider constrained polynomial optimization problems, that is the minimization or maximization of a polynomial over a given set. The most general sets that we will take into consideration in this work are semialgebraic sets.

A *semialgebraic set* is a subset of $\mathbb{R}^n$ defined by a boolean combination of polynomial inequalities. Throughout this work, we will implicitly assume that all semialgebraic sets are *basic closed semialgebraic set*, that means that they are the solution set of a finite system of non-strict polynomial inequalities. Every basic closed semialgebraic set $K$ has a representation of the form

$$K = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0,\ g_i \in \mathbb{R}[x],\ i \in [k]\} \subset \mathbb{R}^n. \tag{2.18}$$

We will oftentimes use $G$ as a shorthand for the set of constraints, that is $G := \{g_1, \ldots, g_k\}$.

It is not hard to see that every polyhedron is a semialgebraic set, as it is defined by non-strict linear inequalities. In the same spirit, we can see that every spectrahedron is a semialgebraic set, since it can be described as the positivity region of the principal minors of the linear matrix pencil that defines the spectrahedron.

We define a *polynomial optimization problem* as the problem of minimizing a polynomial over a semialgebraic set

$$\begin{aligned} \text{minimize} \quad & p(x) \\ \text{s.t.} \quad & x \in K \end{aligned} \tag{2.19}$$

with optimal value $p_{min,K}$.

With the remark above it becomes apparent that every linear and every semidefinite programming problem is a polynomial optimization problem. However, the class of polynomial optimization problems is by far more general. It incorporates a large number of convex and non-convex problems. For example binary linear programming problems can be encoded by adding constraints of the form $x_i - x_i^2 = 0$ to the formulation. Many NP-hard problems can be expressed in this form.

As in the unconstrained case, we approach the problem by first rewriting it in what is known as the dual form.

$$\begin{aligned} \text{maximize} \quad & t \\ \text{s.t.} \quad & p(x) - t \geq 0, \\ & x \in K. \end{aligned} \tag{2.20}$$

In the same manner as we used sum of squares polynomials in the unconstrained setting, we encode the condition $p(x)-t \geq 0$ on $K$ using Positivstellensätze. These representation results from real algebraic geometry allow to encode the problem as infinite dimensional linear or semidefinite programs. By restricting the degree of the polynomials in the representation, we get hierarchies of finite dimensional linear or semidefinite programming relaxations.

We will now state some well-known Positivstellensätze. We begin with a celebrated result by Polyá that provides a certificate of positivity for a homogeneous polynomial on the simplex.

**Theorem 2.12** (Polyá [72]). *Let $p \in \mathbb{R}[x]$ be a homogeneous polynomial. If $p > 0$ on the simplex $\{x \in \mathbb{R}^n_+ \mid \sum_{i=1}^n = 1\}$, then for sufficiently large $d \in \mathbb{N}$, all coefficients of the polynomial $(\sum_{i=1}^n x_i)^d p$ are non-negative.*

In a more general setting, namely for compact semialgebraic sets, we can use a Positivstellensatz due to Schmüdgen. Remember that $[1]_0^k$ stands for all $k$-dimensional 0/1-vectors as introduced in Section 2.1.

**Theorem 2.13** (Schmüdgen [81, Corollary 3]). *Assume the semialgebraic set $K$ is compact. Given $p \in \mathbb{R}[x]$, if $p > 0$ on $K$, then $p = \sum_{\alpha \in [1]_0^k} \sigma_\alpha g^\alpha$ with $\sigma_\alpha \in \Sigma$.*

By applying Schmüdgen's Positivstellensatz to the constrained optimization problem (2.20) and restricting the degree of the representation, we get the following problem. It is in fact a semidefinite programming problem, since we have seen that sums of squares can be recognized by semidefinite programming.

$$\begin{aligned} \text{maximize} \quad & t \\ \text{s.t.} \quad & p(x) - t = \sum_{\alpha \in [1]_0^k} \sigma_\alpha g^\alpha, \\ & \sigma_\alpha \in \Sigma,\ deg(\sigma_\alpha g^\alpha) \leq d \end{aligned} \tag{2.21}$$

For a set of constraint polynomials $G = \{g_1, \ldots, g_k\}$, we denote by $p^{(d)}_{Sch,G}$, the optimal value of the relaxation with degree bounded by $d$. This semidefinite program can grow very large, since it involves $2^k$ sums of squares to be checked. This is the reason why for most practical purposes the Positivstellensatz of Putinar comes into application. Its use was proposed by Lasserre, he describes a dual approach using moment matrices in [46]. The relaxation method based on Putinar's Positivstellensatz, which we describe below has hereupon become known as Lasserre relaxation. Before we can state Putinar's Positivstellensatz, we need some more notation.

**Definition 2.14.** Given polynomials $g_1, \ldots, g_k \in \mathbb{R}[x]$, the set $M(g_1, \ldots, g_k) := \{\sigma_0 + \sum_{i=1}^k \sigma_i g_i \mid \sigma_0, \sigma_i \in \Sigma\}$ is called the *quadratic module* generated by $g_1, \ldots, g_k$.

A *truncated quadratic module* $M_d(g_1, \ldots g_k)$ is the subset of $M(g_1, \ldots, g_k)$ that contains all polynomials up to degree $d$, that is $M_d(g_1, \ldots g_k) := \{\sigma_0 + \sum_{i=1}^k \sigma_i g_i \mid \sigma_0, \sigma_i \in \Sigma,\ deg(\sigma_0) \leq d,\ deg(\sigma_i g_i) \leq d\}$.

The quadratic module $M(g_1, \ldots, g_k)$ is called *Archimedean* if there exists a polynomial $g \in M(g_1, \ldots, g_k)$ such that the superlevel set $\{x \in \mathbb{R}^n \mid g(x) \geq 0\}$ is compact.

Now we are ready to state the Positivstellensatz.

**Theorem 2.15** (Putinar's Positivstellensatz [75, Theorem 1.4]). *Let $K$ be a semialgebraic set as in 2.18. Assume that the quadratic module $M(g_1, \ldots, g_k)$ is Archimedean. If $p > 0$ on $K$, then $p \in M(g_1, \ldots, g_k)$.*

Now we can apply Putinar's Positivstellensatz to optimization problem (2.20). By restricting degrees we end up with a different hierarchy of semidefinite programs. Unlike in (2.21), only $k$ sums of squares terms are involved, which means a significant reduction in complexity for large numbers of inequalities. We get the following semidefinite program

$$\begin{aligned} &\text{maximize} \quad t \\ &\text{s.t.} \quad p(x) - t = \sigma_0 + \sum_{i=1}^{k} \sigma_i g_i, \\ &\qquad \sigma_i \in \Sigma, \ deg(\sigma_i g_i) \leq d. \end{aligned} \tag{2.22}$$

Its optimal value is denoted by $p^{(d)}_{Put,G}$.

The representation results by Schmüdgen and by Putinar both use sums of squares and the resulting relaxations are semidefinite programs. Remarkably, there also exists a result due to Handelman that leads to linear programming relaxations in the special case when $K$ is a polytope.

**Theorem 2.16** (Handelman, [26, Proposition I.1]). *Let $p \in \mathbb{R}[x]$ and let $P$ be a polytope described by linear inequalities $g_1(x) \geq 0, \ldots g_k(x) \geq 0$. If $p$ is positive on $P$, then it admits a representation*

$$p(x) = \sum_{\alpha \in \mathbb{N}^k} \lambda_\alpha g^\alpha, \lambda_\alpha \geq 0.$$

Recently, Averkov published a constructive proof of this theorem, see [5].

A more general version of this Positivstellensatz was rediscovered several times. Prestel and Delzell present a good overview in their book [74]. It generalizes Handelman's result to nonlinear constraints.

**Theorem 2.17** ([53, Theorem 3.30]). *Assume $K$ is compact and the polynomials $g_1, \ldots, g_k$ satisfy $0 \leq g_i \leq 1$ on $K$ for all $i$. Furthermore, together with 1, they generate the algebra $\mathbb{R}[x]$, i.e., $\mathbb{R}[x] = \mathbb{R}[1, g_1, \ldots, g_m]$. If $p$ is positive on $K$, then it admits a representation of the form*

$$p = \sum_{\alpha, \beta \in \mathbb{N}^k} c_{\alpha\beta} g^\alpha (1-g)^\beta$$

*for finitely many non-negative scalars $c_{\alpha\beta}$.*

Since we will focus on relaxations derived by Handelman's Positivstellensatz in the following, we introduce some more notation. The Positivstellensatz motivates the following definition.

**Definition 2.18** (Handelman set)**.** Given $d \in \mathbb{N}$ and a set of polynomials $G = \{g_1, \ldots, g_k\}$ define the *Handelman set* (or the *preprime*) generated by $G$ as

$$H_d(G) := \left\{ \sum_{\alpha \in \mathbb{N}^n} \lambda_\alpha g^\alpha | \quad \lambda_\alpha \geq 0, \quad deg(g^\alpha) \leq d \right\}. \tag{2.23}$$

**Lemma 2.19.**

*Let $f \in H_{d_f}(G),\ g \in H_{d_g}(G)$, then*
1. *$fg \in H_{d_f+d_g}(G)$,*
2. *$f + g \in H_{\max\{d_f,d_g\}}(G)$,*

*Let $G = \{h_i,\ i = 1, \ldots, k,\ h_i \in \mathbb{R}[x]_1\}$ and $\nu_i \geq 0$, then $(\sum_{i=1}^k \nu_i h_i)^r \in H_r(G)$.*

*Proof.* The statements can easily be seen by expanding the Handelman-representation. □

As for the other representation results, Handelman's result can be applied to the polynomial optimization problem (2.20), leading to the following hierarchy of relaxations.

$$\begin{aligned} &\text{maximize } t \\ &\quad \text{s.t. } p(x) - t \in H_d(G). \end{aligned} \tag{2.24}$$

The optimal value of this problem is denoted by $p^{(d)}_{Han,G}$.

### 2.3.3 Polynomial Optimization over Spectrahedra

Problems involving a polynomial objective function and positive semidefinite constraints on matrix polynomials are called *polynomial matrix inequality* (PMI) problems and can be written in the following standard form.

$$\begin{aligned} \text{minimize } &\ p(x) \\ \text{s.t. } &\ G(x) \succeq 0, \end{aligned} \tag{2.25}$$

where $p(x) \in \mathbb{R}[x]$ and $G(x) \in \mathcal{S}_k[x]$, not necessarily linear, for $x = (x_1, \ldots, x_n)$. Remember that the matrix inequality condition defines a semialgebraic set, since the condition can be translated to a condition on the principal minors of the matrix $G(x)$. From this point of view, problem (2.25) is a polynomial optimization problem with $2^k$ inequalities and the methods from the last section can be applied. However the large number of inequalities already for small $k$ make this approach disadvantageous.

Alternatively, Lasserre proposes to use an extension of Descartes' rule of signs to rewrite the positive semidefiniteness condition (described in [48], as cited in [31]). In this approach, we are left with only $n$ inequalities stemming from the coefficients of the characteristic polynomial of $G(x)$. This approach, too, has a significant drawback. It leads to complex Positivstellensatz representations involving polynomials of high degrees.

To circumvent this problem Hol and Scherer [33], and Kojima [43] introduced sums of squares relaxations for PMIs leading to semidefinite programming relaxations of the original problem and extending the results from the last section. This leads to relaxations that use a smaller number of variables and polynomials of smaller degrees.

Analogously to the scalar case, they introduce sums of squares multipliers to define a representation of the polynomial $p(x)$.

**Definition 2.20.** A symmetric matrix-valued $k \times k$ polynomial matrix $S(x)$ is said to be a sum of squares matrix if there exists a (not necessarily square) polynomial matrix $T(x)$ such that $S(x) = T(x)^T T(x)$. We denote the set of all $k \times k$ sum of squares matrices in $n$ variables by $\Sigma_n^{k\times k}$, and by $\Sigma_{n,d}^{k\times k}$ the subset of $\Sigma_n^{k\times k}$ where each entry has degree at most $d$.

Note that for $k = 1$, the definition coincides with the definition of sum of squares polynomials given earlier.

In a similar fashion to the Positivstellensatz relaxations from the last section, Hol and Scherer propose to solve the following optimization problem.

$$\begin{aligned} \text{maximize} \quad & t \\ \text{s.t.} \quad & p(x) - \langle S(x), G(x)\rangle - t \ \in \Sigma_n \\ & S(x) \ \in \Sigma_n^{k\times k}. \end{aligned} \tag{2.26}$$

The following theorem shows that under mild assumptions, the optimal values of problems (2.25) and (2.26) coincide. This constraint qualification is similar to the assumptions of an Archimedean quadratic module in Putinar's Positivstellensatz and, in fact, derived from there. It ensures that the optimal value of the hierarchy converges to the optimal value of the original optimization problem (2.25) for growing bound on the degree $d$.

**Theorem 2.21** ([33, Theorem 1])**.** *If there exists some $r > 0$ and some SOS matrix $R(x)$ such that*

$$r - \|x\|^2 - \langle R(x), G(x)\rangle \in \Sigma_n,$$

*then the optimal values of* (2.25) *and of* (2.26) *are equal.*

To make the optimization problem (2.26) computationally tractable, the degree of the SOS matrix $S(x)$ can be restricted. For each degreebound $d$ we attain a problem of the following form.

$$\begin{aligned} \text{maximize} \quad & t \\ \text{s.t.} \quad & p(x) - \langle S(x), G(x) \rangle - t \ \in \Sigma_n \\ & S(x) \ \in \Sigma_{n,d}^{k \times k}. \end{aligned} \tag{2.27}$$

Problem (2.27) is a semidefinite program. To see this, we will now explain how to check whether a given matrix is a sum of squares matrix. We need a generalized version of the trace product. If $M \in \mathbb{R}^{nk \times nk}$ is partitioned into $n \times n$ blocks as $(M_{ij})_{i,j=1}^{k}$, define

$$\mathrm{Tr}_k(M) = \begin{pmatrix} \mathrm{Tr}(M_{11}) & \cdots & Tr(M_{1k}) \\ \vdots & \ddots & \vdots \\ \mathrm{Tr}(M_{k1}) & \cdots & \mathrm{Tr}(M_{kk}) \end{pmatrix}$$

and the bilinear mapping $\langle ., . \rangle : \mathbb{R}^{kk \times nk} \times \mathbb{R}^{kk \times nk} \to \mathbb{R}^{k \times k}$ as

$$\langle A, B \rangle_k = \mathrm{Tr}_k(A^T B).$$

Now we are ready to state the representation result.

**Theorem 2.22** ([33, Lemma 2])**.** *Let $u(x)$ be a polynomial vector whose components $u_j(x)$, $j \in [n_u]$, contain all pairwise different monomials in the variables $x_1 \dots x_n$ up to a certain degree $d$, and denote the pairwise different monomials in $u(x)u(x)^T$ by $w_j(x)$, $j \in [n_w]$. Then we can uniquely describe $u(x)u(x)^T$ in terms of symmetric matrices $Z_j$, such that*

$$u(x)u(x)^T = \sum_{j=1}^{n_w} Z_j w_j(x).$$

*The matrix polynomial $S(x)$ is SOS with respect to the monomial basis $u(x)$ if and only if there exist symmetric matrices $S_j$ such that $S(x) = \sum_{j=1}^{n_w} S_j w_j(x)$, and the linear system*

$$\langle W, I_k \otimes Z_j \rangle_k = S_j, \ j \in [n_w]$$

*has a solution $W \succeq 0$.*

An alternative treatment of polynomial matrix inequality problems is given in [31]. Lasserre and Henrion focus mainly on the dual viewpoint of moment relaxations. As in Lasserre's moment method for polynomial optimization [46], the basic idea is to linearize all polynomials by introducing a new variable for each monomial. The relations among the monomials give semidefinite conditions on the moment matrices.

## 2.4 Properties of Positivstellensatz Relaxations

All representation results shown in the last section serve the same purpose, to certify the positivity of a polynomial over a given set.

As soon as the practical application in polynomial optimization comes into play, a number of factors have to be considered when choosing a Positivstellensatz. This section aims to provide a high level comparison of the Positivstellensätze of Schmüdgen, Putinar and Handelman and to motivate why we focus on Handelman's representation subsequently. Here, we will consider truncated representations, that is, representations that involve only monomials up to a certain degree. The results introduced in the last section show that (as long as the conditions are fulfilled) every polynomial does have a corresponding representation for unbounded degrees, but we do not gain any information on the convergence or error bounds for the truncated situation.

When comparing truncated representations, Schmüdgen's result is clearly the result with the highest expressive power. Not only are the preconditions the weakest, also every Putinar and every Handelman representation is a representation of Schmüdgen type. The first statement is obvious, the second also if we remember that higher exponents of the constraints in the $g^\alpha$ term of the Handelman representation can be pulled into the sums of squares multipliers of the Schmüdgen representation. This stronger expressiveness comes, as already hinted in the last section, at cost of higher complexity in the representation. Schmüdgen's representations involves $2^k$ terms, while $k+1$ terms suffice for Putinar's representation.

A direct comparison of the Handelman and Putinar result is harder. Situations in which one of the two is stronger are known, as we will see in the following examples.

**Example 2.23** ([13, p. 15])**.** Let $K$ be the 0/1-hypercube as defined in (2.9). The monomial $\prod_{i=1}^n x_i \subset \mathbb{R}^n$ belongs to the Handelman set spanned by the inequalities of the hypercube $H_t(x_i, 1-x_i)$ for any $t \geq n$, but does not belong to the quadratic module $M_t(x_i, 1-x_i)$. As De Klerk and Laurent remark, it does however belong to the quadratic module after adding a suitable constant $C_n \leq 1$ for even $n$: $\prod_{i=1}^s x_i + C_n \in M_n(x_i, 1-x_i)$.

**Example 2.24** ([51, Example 5.5])**.** Let $K = [0,1]$. The polynomial $p(x) = (x - \frac{1}{2})^2$ belongs to $M_2(x, 1-x)$ since it is a square.

It does only belong to $H_2(x, 1-x)$ after adding $\frac{1}{4}$ to get the representation $p(x) + \frac{1}{4} = \frac{1}{2}(1-x)^2 + \frac{1}{2}x^2$.

It is also interesting to study how fast the respective approaches converge. Asymptotic convergence follows directly from the representation results, that is we have

$$\lim_{d\to\infty} p_{Han}^d = \lim_{d\to\infty} p_{Put}^d = \lim_{d\to\infty} p_{Sch}^d = p_{min}.$$

But more can be said about convergence. Due to its importance in practice, the behavior of the Putinar approach is well studied. There exists a range of results on

cases, in which Putinar's approach converges after finitely many steps. For convex problems we have the following.

**Theorem 2.25** ([14, Corollary 3.3])**.** *Consider problem* (2.19) *under the following assumptions:*

1. *The polynomials $p, g_1, \ldots, g_k$ are convex;*
2. *The Slater condition holds:*

$$\begin{aligned} \exists x_0 \in \mathbb{R}^n \ &: g_i(x_0) > 0 \text{ for } g_i \text{ nonlinear and} \\ &\quad g_i(x_0) \geq 0 \text{ for } g_i \text{ linear;} \end{aligned}$$

3. *The quadratic module $M(g_1, \ldots, g_k)$ is Archimedean;*
4. *$\bigtriangledown^2 p(x^*) \succ 0$(i.e. the Hessian of $p$ at $x^*$ is positive definite) if $x^*$ is a minimizer of* (2.19).

*Then one has finite convergence of the Putinar approach.*

Indeed, in practical computations, finite convergence can oftentimes be observed. A fact that is supported by the following result by Nie. It states that, generically, finite convergence can be expected.

**Theorem 2.26** ([67, Theorems 1.1 and 1.2])**.** *Suppose the Archimedean condition holds for the quadratic module $M(g_1, \ldots g_k)$. Let $d_0, d_1, \ldots, d_m$ be positive integers. Then there exist a finite set of polynomials $\Phi_1, \ldots, \Phi_L$, which are in the coefficients of polynomials $f \in R[x]_{d_0}, g_j \in R[x]_{d_j}$ for $j \in [m]$, such that if $\Phi_1, \ldots, \Phi_L$ do not vanish at the input polynomial, then the Putinar approach* (2.22) *has finite convergence.*

Unfortunately, similarly strong results are not known for Handelman's approach. Quite the contrary is the case, as Lasserre explains in [51, p.126]. He gives a simple argument why the convergence can never be finite, if the optimal point lies in the interior of $K$. The argument is as follows.

Assume finite convergence for Handelman's approach on a polytope $K$, that is $p_{min} = p^d_{Han}$ for some $d \in \mathbb{N}$. The interior of $K$ is given by $intK := \{x \in \mathbb{R}^n \mid g_j(x) > 0,\ j \in [k]\}$. If an optimal point $x^*$ lies in $intK$, we get the contradiction that

$$0 = p_{min} - p^d_{Han} = \sum_{\substack{\alpha \in \mathbb{N}^n \\ deg(g^\alpha) \leq d}} \lambda_\alpha g^\alpha(x^*) > 0.$$

This means that potentially high relaxation orders have to be chosen to get good results. Lasserre notes in [47, p.388] that this may cause numerical problems due to large binomial coefficients in the expansion of terms $g_j^{\alpha_j}$ for large exponents $\alpha_j$.

It may still be favorable to choose the approach using Handelman's representation result. The fact that the resulting relaxation is a linear programming problem instead of a semidefinite programming problem for approaches using Schmüdgen or Putinar representations cannot be underestimated. While semidefinite programming receives a lot of interest due to broad applications, solvers are still not as

mature as solvers for linear programming. Only recently, first industrial codes (MOSEK) appeared. Still semidefinite programming is only practical for problems with a small to moderate number of variables, (see also the ongoing benchmarking project [64]).

Linear programming solvers on the other hand can handle problems with millions of variables and constraints (as claimed by IBM's CPLEX in advertisements). Attempts to strengthen the linear relaxations but still keep the nice computational behavior have been made recently (see e.g. [52]). Our results in Section 4 also point in this direction.

The Handelman approach is also attractive from a theoretical point of view. The linear structure of the representation allows stricter analysis concerning error bounds and degree bounds for representations. We will focus on these topics in the next section.

# 3 Error Bounds for Handelman Approximations

In the last chapter, we introduced several hierarchical methods to relax polynomial optimization problems. The quality of such a relaxation can be assessed with error and degree bounds.

Our main contribution in this chapter is to generalize the error bound for the Handelman representation for polynomial optimization over the unit cube to arbitrary boxes and even further to polytopes. Not very surprisingly, the bound also depends on a factor coming from the representation of the feasibility region. Unlike the known results for the Putinar and Schmüdgen case, this constant can be described in our case. It is simply the longest edge of a bounding box of the polytope $K$.

We will start by reviewing previously known bounds for the Positivstellensätze by Putinar and Schmüdgen in section 3.1. In this section we also state a bound given by De Klerk and Laurent [13] for the Handelman approach in the special case where the constraints define the unit cube. It is this result, which is generalized in the following sections. In principle this can be achieved by a coordinate transformation mapping a given box to the unit cube. We make this computation very explicit, allowing us to derive a bound depending on the length of the edges of the cube. This result is the key to show convergence of a branch and bound scheme that uses the Handelman Positivstellensatz for the bounding in the next chapter.

To derive our bound, we make use of the similarity of the Handelman representation on the box with Bernstein polynomials. After recapping some basics on Bernstein polynomials in Section 3.2, we describe the ideas from [13] on how to use the Bernstein polynomials to find bounds on the Handelman representation in Section 3.3.

Bernstein polynomials have a long history for their use in bounding interval polynomials. The "Bernstein algorithm" is a standard tool in polynomial optimization. Error bounds for the Bernstein bound for polynomials are known to the community. We review these bounds thoroughly and extend them to an error bound for the Handelman approach. We introduce ideas in the univariate case in Section 3.4 and readily generalize the concept to the multivariate setting in Section 3.5.

Section 3.6 is concerned with the connection of error and degree bounds. It is a common procedure to extract degree bounds from known error bounds and we will perform this task using the error bounds derived before.

Finally, in Section 3.7, we will further explore the similarities and differences

between the Bernstein approximation and the Handelman relaxation.

To improve readability, we omit some lengthy calculations in the proofs. The complete proofs are collected in Section 3.8.

## 3.1 Previously Known Bounds

Most error and degree bounds for Positivstellensatz relaxations depend on the following three parameters.

1. A constant $c$, which depends on the description of the set $K$ over which to optimize;
2. $\deg(p)$, the degree of the objective polynomial $p$;
3. $\|p\|$ for the error bound and $\frac{\|p\|}{p_{min}}$, a measure of how far $p$ is from having a zero on $K$, for the degree bound.

The norm in the last parameter is defined as

$$\|p\| := \max_{\alpha} |p_\alpha| \frac{\prod_{j=1}^{n} \alpha_j!}{(\sum_{j=1}^{n} \alpha_j)!}.$$

Note that this does indeed define a norm on the real vector space $\mathbb{R}[x]$, measuring the size of the coefficients of a polynomial.

The first result that we will state is concerned with Pólya's theorem 2.12. Powers and Reznick were able to prove a degree bound using only elementary methods. Many later results, also such that are concerned with other representations, depend on the following theorem.

**Theorem 3.1** ([73, Theorem 1])**.** *Let $p \in \mathbb{R}[x]$ be a homogeneous polynomial of degree $d$. If*

$$N > \frac{d(d-1)}{2} \frac{\|p\|}{p_{min}} - d,$$

*then $(x_1 + \ldots + x_n)^N f(x_1, \ldots, x_n)$ has positive coefficients.*

Deriving bounds for Schmüdgen's Positivstellensatz is a more complex task. An early result by Prestel states that a degree bound depending on the three parameters mentioned at the beginning of the section does exist. A (nearly) concrete formulation of the bound has been found by Schweighofer. The bound, which we will state in the next theorem, depends on a constant $c$, which in theory can be computed from the representation of the set $K$. This task, however, is a complicated undertaking and, as Schweighofer and Nie state in [69], "probably too tedious" to perform in practice. The proof makes use of the effective version of Pólya's theorem stated above.

**Theorem 3.2** ([84, Theorem 3, Theorem 4])**.** *Let $K$ be as in (2.18) and assume $K \subset (-1, 1)^n$. Then there are some $c \in \mathbb{N}$ and $1 < c' \in \mathbb{N}$ with the following properties.*

*Every $p \in \mathbb{R}[x]$ of degree $m$ with $p_{min,K} > 0$ has a representation of the form $p = \sum_{\alpha \in [1]_0^k} \sigma_\alpha g^\alpha$, where $\sigma_\alpha \in \Sigma[x]$ is such that*

$$\sigma_\alpha = 0 \quad or \quad \deg(\sigma_\alpha g^\alpha) \leq cm^2 \left(1 + \left(m^2 n^m \frac{\|p\|}{p_{min,K}}\right)^c\right)$$

*for all $\alpha \in [1]_0^k$.*

*For every polynomial $p$ of degree $m$ and for all integers $d \in \mathbb{N}$, with $d \geq c' m^{c'} n^{c'm}$, we have*

$$p_{min,K} - p_{Sch,G}^{(d)} \leq c' m^4 n^{2m} \frac{\|p\|}{\sqrt[c']{d}}.$$

From this result, Nie and Schweighofer derived a bound for Putinar's Positivstellensatz. Due to missing mixed terms in the representation, the result is weaker and involves an exponential term. However, the bound is not expected to be sharp. Nie and Schweighofer even speculate whether it can be reduced to the bound given above for the Schmüdgen case.

**Theorem 3.3** ([69, Theorem 6, Theorem 8])**.** *Let $K$ be as in (2.18) and assume $K \subset (-1,1)^n$. Let the quadratic module $M(g_1, \dots g_k)$ generated by the polynomials defining $K$ be Archimedean. Then there are some $c \in \mathbb{R}_+$ and $1 < c' \in \mathbb{N}$ with the following properties.*

*Every $p \in \mathbb{R}[x]$ of degree $m$ with $p_{min,K} > 0$ is in $M_d(g_1, \dots, g_k)$, where*

$$d \leq c \exp\left(\left(m^2 n^m \frac{\|p\|}{p_{min}}\right)^c\right).$$

*For every polynomial $p$ of degree $m$ and for all integers $d \in \mathbb{N}$, satisfying $d \geq c \exp\left(\left(2m^2 n^m\right)^c\right)$, we have*

$$p_{min,K} - p_{Put,G}^{(d)} \leq \frac{6m^3 n^{2m} \|p\|}{\sqrt[c']{log \frac{d}{c'}}}.$$

It would be interesting to refine these bounds for special sets, for which the unknown constant $c$ can be calculated. For the case of the hypercube, De Klerk and Laurent [13] propose to relate the error bounds for the Schmüdgen case (which we will discuss below) to the Putinar case. Their argument depends on a representation of mixed terms of the form $\prod_{i=1}^n x_i + C_n$ in the quadratic module generated by polynomials $x_i - x_i^2$ describing the hypercube for some constant $C_n$. They conjecture that the smallest possible $C_n$ for such a representation is $C_n = 1/n(n+2)$. The argument is further refined in the recent work by Magron ([60]).

For minimization problems, a natural upper bound for the optimal value can be found by discretizing the problem. The feasibility region is intersected with a regular grid and the objective is evaluated at every point in the intersection. To formalize this, we define $S(d)$ as the set of points in $S$ with denominator $d$

$$S(d) := \{x \in S \mid dx \in \mathbb{N}^n\}. \tag{3.1}$$

For example, $p_{min,[\underline{a},\underline{b}](d)}$ is the minimum of $p(x)$ over the points in

$$[\underline{a},\underline{b}](d) = \left\{ a_1 + \frac{k_1}{d}(b_1 - a_1), \ldots, a_n + \frac{k_n}{d}(b_n - a_n) \mid k \in [d]_0^n \right\}.$$

In [13], de Klerk and Laurent derive a bound for the error of the Handelman approximation on the hypercube $Q = [0,1]^n$. They use the fact that the minimum of a polynomial over the cube $Q$ is bounded from below by the Bernstein approximation introduced in the next section and from above by the minimum over the set $Q(d)$ of points in $Q$ with denominator $d$. We state their main result in the following theorem.

**Theorem 3.4** ([13, Theorem 1.4]). *Let $Q = \{x \in \mathbb{R}^n \mid g_j(x) \geq 0,\ j = 1, \ldots, 2n\}$ be the hypercube defined by the $2n$ linear polynomials $g_1 = x_1,\ g_2 = 1 - x_1, \ldots, g_{2n-1} = x_n,\ , g_{2n} = 1 - x_n$. For $d \geq \deg(p)$ we have*

$$\begin{aligned} \max(p_{min,Q} - p_{Han,G}^{(nd)},\ p_{min,Q(d)} - p_{min,Q}) &\leq \frac{\|p\|)}{d}\binom{\deg(p)+1}{3} n^{\deg(p)} \\ &\leq \frac{\|p\|}{d}\frac{deg(p)^3 n^{\deg(p)}}{6}. \end{aligned}$$

Employing similar ideas as in [13], we will use the Bernstein representation of the objective polynomial to derive a new error bound on the Handelman approximations $p_{Han,g}$. We generalize earlier results on the Bernstein approximation (see [18, 61]) to arbitrary boxes. These results are used to obtain an error bound for the general Handelman approach on polytopes in Theorem 3.15.

## 3.2 Preliminaries on Bernstein Polynomials

Before stating the error bounds, we have to introduce some facts on Bernstein polynomials. The Bernstein polynomial basis was first introduced in 1912 as a tool to constructively proof the Stone-Weierstrass theorem [7]. Since then, it has been applied in many areas, most notably for computer aided design.

Our interest in Bernstein polynomials lies in the similarity to the polynomials appearing in Handelman representations over boxes. In order to produce results for arbitrary boxes, we use a generalized version of the Bernstein polynomials and the Bernstein algorithm, mostly following the notation in [13, 18] and [78].

**Definition 3.5.** Given $a, b \in \mathbb{R},\ a \leq b$, define the generalized univariate Bernstein polynomials of degree $d$ as

$$p_{j,d}(x,a,b) := \binom{d}{j}\frac{(x-a)^j(b-x)^{d-j}}{(b-a)^d}.$$

The (generalized) Bernstein approximation of a polynomial $p(x) \in \mathbb{R}[x]$ of degree $d \geq \deg(p)$ is given by

$$B_d(p,a,b) := \sum_{j=0}^{d} p(a + \frac{j}{d}(b-a))p_{j,d}(x,a,b). \tag{3.2}$$

For the multivariate case, let $\beta \in [n]_0^d$, $a, b \in \mathbb{R}^n$. Analogously to the univariate case, define the multivariate Bernstein polynomials by

$$P_{\beta,d}(x,a,b) := \prod_{i=1}^{n} p_{\beta_i,d}(x_i,a_i,b_i) = \prod_{i=1}^{n} \binom{d}{\beta_i} \frac{(x_i-a_i)^{\beta_i}(b_i-x_i)^{d-\beta_i}}{(b_i-a_i)^d} \tag{3.3}$$

and the multivariate Bernstein approximation by

$$B_d(p,a,b) := \sum_{\beta\in[d]_0^n} p\left(a_1 + \frac{\beta_1}{d}(b_1-a_1), \ldots, a_n + \frac{\beta_n}{d}(b_n-a_n)\right) P_{\beta,d}(x,a,b).$$

These generalized versions inherit some of the nice properties of the usual Bernstein polynomials.

**Lemma 3.6.** *Let $a, b \in \mathbb{R}^n$, $a \le b$. Then the polynomials $P_{\beta,d}(x,a,b)$ are non-negative on the box $[\underline{a}, \underline{b}]$ and the Bernstein approximations of linear polynomials are exact in the following sense:*

$$B_d(1,a,b) = \sum_{\beta\in[d]_0^n} P_{\beta,d}(x,a,b) = 1,$$
$$B_d(x_i,a,b) = \sum_{\beta\in[d]_0^n} \left(a_i + \frac{\beta_i}{d}(b_i-a_i)\right) P_{\beta,d}(x,a,b) = x_i.$$

*Proof.* The polynomials $x-a$ and $b-x$ are non-negative on the interval $[a,b]$. The same is true for their powers and products. The positive combination of these products in $p_{j,d}(x,a,b)$ are thus also non-negative. This easily extends to the multivariate case.

For the exactness of the approximation on linear polynomials, let us first proof the statement for the univariate case.

$$\begin{aligned} B_d(1,a,b) &= \sum_{j=0}^{d} p_{j,d}(x,a,b) = \sum_{j=0}^{d} \binom{d}{j} \frac{(x-a)^j(b-x)^{d-j}}{(b-a)^d} \\ &= \frac{[(x-a)+(b-x)]^d}{(b-a)^d} = \frac{(b-a)^d}{(b-a)^d} = 1, \end{aligned}$$

$$\begin{aligned} B_d(x,a,b) &= a + \sum_{j=0}^{d} \left(\frac{j}{d}(b-a)\right) \binom{d}{j} \frac{(x-a)^j(b-x)^{d-j}}{(b-a)^d} \\ &= a + \frac{(x-a)}{(b-a)^{d-1}} \sum_{j=0}^{d-1} \binom{d-1}{j} (x-a)^j(b-x)^{d-1-j} \\ &= a + (x-a)\sum_{j=0}^{d-1} p_{d-1,j} = x. \end{aligned}$$

The multivariate statement follows by splitting the sum and summing over the single indices $\beta_i \in [d]$ one after another. □

**Lemma 3.7.** *Every Bernstein polynomial $P_{\beta,d-1}(x,a,b)$ of degree $d-1$ can be written as a positive combination of Bernstein polynomials of degree $d$.*

*Proof.* Every univariate Bernstein polynomial $p_{j,d-1}(x,a,b)$ of degree $d-1$ can be written as a positive combination of Bernstein polynomials of degree $d$ as

$$p_{j,d-1}(x,a,b) = \frac{d-j}{d}(b-a)p_{j,d}(x,a,b) + \frac{j+1}{d}(b-a)p_{j+1,d}(x,a,b).$$

See [85, p. 23] for details on the proof in the univariate case on the unit interval. The generalized version is proven analogously by combining the Bernstein basis representations of $(b-x)p_{j,d-1}(x,a,b) = \frac{d-j}{d}(b-a)p_{j,d}(x,a,b)$ and $xp_{j,d-1}(x,a,b) = \frac{j+1}{d}(b-a)p_{j+1,d}(x,a,b)$. For the multivariate version, elevate the degree of the products in the multivariate Bernstein polynomial and apply the distributive law. □

In the following sections, we make use of polynomials with representations in different bases. Let us now give explicit formulas to change between the power basis $1, x_i, x_i^2, x_ix_j, \ldots$, the shifted power basis $1, x_i-a_i, (x_i-a_i)^2, (x_i-a_i)(x_j-a_j), \ldots$, and the Bernstein basis $P_{j,d}(x,a,b)$.

Let $p \in \mathbb{R}[x]_{m,\ldots,m}$ be a polynomial in $n$ variables $x_1,\ldots,x_n$ of degree at most $m$ in each variable and $d \geq m$

$$p(x) = \sum_{\alpha\in[m]_0^n} p_\alpha x^\alpha = \sum_{\gamma\in[m]_0^n} c_\gamma \prod_{j=1}^n (x_j-a_j)^{\gamma_j} = \sum_{\beta\in[d]_0^n} b_\beta^{(d)} P_{\beta,d}(x,a,b).$$

Given the coefficients of the power basis $p_\alpha$, the coefficients of the shifted power basis can be computed by the following formula

$$\forall \gamma \in [m]_0^n,\ c_\gamma = \sum_{\alpha_1=\gamma_1}^m \cdots \sum_{\alpha_n=\gamma_n}^m p_\alpha \prod_{j=1}^n \binom{\alpha_j}{\gamma_j} a_j^{\alpha_j-\gamma_j}. \tag{3.4}$$

This can be seen by expanding the power basis representation as

$$\begin{aligned}
p(x) &= \sum_{\alpha\in[m]_0^n} p_\alpha x^\alpha = \sum_{\alpha\in[m]_0^n} p_\alpha \prod_{j=1}^n ((x_j-a_j)+a_j)^{\alpha_j} \\
&= \sum_{\alpha\in[m]_0^n} p_\alpha \prod_{j=1}^n \left( \sum_{\gamma_j=0}^{\alpha_j} \binom{\alpha_j}{\gamma_j} (x_j-a_j)^{\gamma_j} a_j^{\alpha_j-\gamma_j} \right) \\
&= \sum_{\gamma\in[m]_0^n} \left[ \sum_{\alpha_1=\gamma_1}^m \cdots \sum_{\alpha_n=\gamma_n}^m p_\alpha \prod_{j=1}^n \binom{\alpha_j}{\gamma_j} a_j^{\alpha_j-\gamma_j} \right] \prod_{j=1}^n (x_j-a_j)^{\gamma_j} \\
&= \sum_{\gamma\in[m]_0^n} c_\gamma \prod_{j=1}^n (x_j-a_j)^{\gamma_j}.
\end{aligned}$$

To derive a formula for the coefficients in the Bernstein basis in terms of the coefficients of the shifted power basis, let us take a look at the Bernstein coefficients of a univariate shifted monomial $(x-a)^s$.

**Lemma 3.8** ([78, p. 228])**.** *For $s \in [d]$,*

$$(x-a)^s = \sum_{j=s}^{d} \frac{\binom{j}{s}}{\binom{d}{s}} (b-a)^s p_{j,d}(x,a,b).$$

*The Bernstein polynomials $P_{1,d}, \ldots, P_{d,d}$ form a basis of the vectorspace of polynomials of degree at most $d$.*

*Proof.*

$$\begin{aligned}
(x-a)^s &= (x-a)^s (x-a+b-x)^{d-s} \frac{1}{(b-a)^{d-s}} \\
&= \sum_{j=0}^{d-s} \binom{d-s}{j} \frac{1}{(b-a)^{d-s}} (x-a)^{s+j} (b-x)^{d-s-j} \\
&= \sum_{j=s}^{d} \binom{d-s}{j-s} \frac{(x-a)^i (b-x)^{d-s}}{(b-a)^{d-s}} \\
&= \sum_{j=s}^{d} \frac{\binom{j}{s}}{\binom{d}{s}} \binom{d}{j} \frac{(x-a)^j (b-x)^{d-j}}{(b-a)^{d-s}}.
\end{aligned}$$

The last equality follows from

$$\binom{d-s}{j-s} = \frac{(d-s)!}{(j-s)!(d-j)!} = \frac{\left(\frac{j!}{s!(j-s)!}\right)}{\left(\frac{d!}{s!(d-s)!}\right)} \frac{d!}{j!(d-j)!} = \frac{\binom{j}{s}}{\binom{d}{s}} \binom{d}{j}.$$

□

It follows directly, that

$$\begin{aligned}
\prod_{j=1}^{n} (x_j - a_j)^{\gamma_j} &= \sum_{\beta_1=\gamma_1}^{d} \cdots \sum_{\beta_n=\gamma_n}^{d} \left( \prod_{j=1}^{n} \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} (b_j - a_j)^{\gamma_j} P_{\beta,d}(x,a,b) \right) \\
&= \sum_{\substack{\beta \in [d]_0^n \\ \beta \geq \gamma}} \left( \prod_{j=1}^{n} \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} (b_j - a_j)^{\gamma_j} P_{\beta,d}(x,a,b) \right).
\end{aligned}$$

Now we can express the Bernstein coefficients in terms of the coefficients of the shifted power basis

$$b_\beta^{(d)} = \left( \sum_{\gamma_1=0}^{\beta_1} \cdots \sum_{\gamma_n=0}^{\beta_n} c_\beta \prod_{j=1}^{n} \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} (b_j - a_j)^{\gamma_j} \right). \tag{3.5}$$

*Remark* 3.9. Bernstein polynomials can also be defined on a simplex instead of the hyperrectangular domain used here. Error bounds for this setting have been studied by Leroy [55, 56]. We focus on the version defined on hyperrectangular sets, because it fits nicely with the Handelman relaxations on the box, as we will see in the next section.

## 3.3 Using Bernstein Bounds to Bound the Handelman Approximation

In this section, we show how a Bernstein representation of the objective polynomial can be used to bound the optimal value of the Handelman approximation. The ideas from [13, Section 1.3] on Bernstein approximations on the unit cube directly translate to the setting with arbitrary boxes.

Let $p^{(d)}_{Ber(a,b)} := \min_{\beta\in[d]^n_0} b^{(d)}_\beta$ be the smallest coefficient of the representation of $p$ in the generalized Bernstein basis (3.3). Analogously to the case, where $a = 0$ and $b = 1$, as discussed in [13], we use Lemma 3.6 to see that

$$p(x) - p^{(d)}_{Ber(a,b)} = \sum_{\beta\in[d]^n_0} \left(b^{(d)}_\beta - p^{(d)}_{Ber(a,b)}\right) P_{\beta,d}(x,a,b).$$

$P_{\beta,d}(x,a,b)$ is non-negative on $[\underline{a},\underline{b}]$ (Lemma 3.6) and $\left(b^{(d)}_\beta - p^{(d)}_{Ber(a,b)}\right)$ is non-negative on $[\underline{a},\underline{b}]$ by definition of $p^{(d)}_{Ber(a,b)}$. Therefore, the left hand side of the equation above is non-negative on $[\underline{a},\underline{b}]$ and the following inequality is true:

$$p^{(d)}_{Ber(a,b)} \leq p_{min,[\underline{a},\underline{b}]}.$$

Again, let $G$ be the set of polynomials defining the box, that means $G = \{x_1 - a_1, \ldots, x_n - a_n, b_1 - x_1, \ldots, b_n - x_n\}$. Since $P_{\beta,d}(x,a,b) \in H_{dn}(G)$, we know that $p(x) - p^{(d)}_{Ber(a,b)} \in H_{dn}(G)$ and thus

$$p^{(d)}_{Ber(a,b)} \leq p^{(dn)}_{Han,G}.$$

Combining the inequalities, we have

$$p^{(d)}_{Ber(a,b)} \leq p^{(dn)}_{Han,G} \leq p_{min,[\underline{a},\underline{b}]} \leq p_{min,[\underline{a},\underline{b}](d)}. \tag{3.6}$$

In the following section, we are going to establish a bound for $p_{min,[\underline{a},\underline{b}](d)} - p^{(d)}_{Ber(a,b)}$ in the univariate case. Remarkably, the bound depends on the length of the interval $[a,b]$. This will be generalized to the multivariate case in Section 3.5.

## 3.4 An Error Bound for Univariate Handelman Approximations

In this section we derive an error bound for the Handelman approximation only in the univariate case, in order to keep the notation simple and to introduce concepts. The results are readily generalized to the multivariate setting in the next section.

The basic ideas for the univariate setting are already contained in Rivlin's work [77] and have been generalized to arbitrary intervals by Rokne in [78].

We are going to derive a bound on $p_{min,[a,b](d)} - p^{(d)}_{Ber(a,b)}$ by bounding the error that is made by the Bernstein approximation (3.2). Given $0 \leq s \leq n$, let us rewrite the polynomial $B_d((x-a)^s, a, b) - (x-a)^s$ in the Bernstein basis with coefficients $\delta_j(s)$.

$$B_d((x-a)^s, a, b) - (x-a)^s = \sum_{j=0}^{d} \delta_j(s) p_{j,d}(x, a, b). \tag{3.7}$$

Next, we will provide a bound on the coefficients $\delta_j(s)$, which we use to deduce a bound on general polynomials afterwards.

**Theorem 3.10** ([78, Theorem 2], see also [77, Theorem 3])**.** *For $d \geq m > 1$, the following holds:*

$$\forall j \in [d]_0, s \in [m]_0 : \ \delta_j(s) \leq \frac{(s-1)^2}{d}(b-a)^s.$$

*Proof.* See Section 3.8. □

With the next theorem, we can relate the Bernstein approximation to the approximation on a regular grid introduced in (3.1).

**Theorem 3.11** ([18, Theorem 4], [61, Theorem 2] and [78, Theorem 3])**.** *For $p(x) = \sum_{i=1}^{m} c_i (x-a)^i$ and $d \geq m \geq 1$, we have*

$$|p_{min,[a,b](d)} - p^{(d)}_{Ber(a,b)}| \leq \Delta(d)$$
$$\text{with} \quad \Delta(d) \leq \vartheta_p \max(|a|^m, 1) \sum_{s=2}^{m} (s-1)^2 \frac{(b-a)^s}{d}$$
$$\text{and} \quad \vartheta_p = \max_{2 \leq s \leq m} \sum_{j=s}^{m} \binom{j}{s} |p_s|.$$

*Proof.* See Section 3.8. □

Using the insights from the last section on the relation of the bounds (3.6), we can now deduce an error bound for the Handelman approximation on arbitrary intervals.

**Corollary 3.12.** *Given an interval $[a,b]$, for any integer $d \geq n$, we have $p_{min,[a,b]} - p^{(dn)}_{Han,G} \leq p_{min,[a,b]} - p^{(d)}_{Ber} \leq \Delta(d)$, where for $|b-a| < 1$,*

$$\Delta(d) \leq \vartheta_p \max(1, a^m)|b-a|_\infty^2 \sum_{s=2}^{m} \left( \frac{(s-1)^2}{d} \right)$$

*and for $|b-a| \geq 1$,*

$$\Delta(d) \leq \vartheta_p \max(1, a^m)|b-a|^m \sum_{s=2}^{m} \left( \frac{(s-1)^2}{d} \right)$$

## 3.5 An Error Bound for Multivariate Handelman Approximations

Error bounds for the Bernstein approach in the fashion of our Theorem 3.14 have been known for a long time now. They seem to be mentioned first by Garloff in [18], the proof however discusses only rectangles in the bivariate setting. Similar results can also be found in [61], again without proof. We will recap all necessary steps and rigorously state the proofs of the error bounds. Most ideas from the univariate setting can be generalized directly to the multivariate setting.

Our main contribution in this section is the generalization of the bounds to the Handelman approach in Theorem 3.15.

Let us look again at the error that is made by the Bernstein approximation of a shifted monomial $B_d(\prod_{j=1}^n (x_j - a_j)^{\gamma_j}, a, b) - \prod_{j=1}^n (x_j - a_j)^{\gamma_j}$, where $d \geq \gamma_j$ for all $j$. We express the error in the Bernstein basis and call the coefficients of the error term $\delta_\beta(\gamma)$.

$$B_d(\prod_{j=1}^{n}(x_j - a_j)^{\gamma_j}, a, b) - \prod_{j=1}^{n}(x_j - a_j)^{\gamma_j} = \sum_{\beta \in [d]_0^n} \delta_\beta(\gamma) P_{\beta,d}(x, a, b)$$

**Theorem 3.13** ([18, Theorem 3])**.** *For $d \geq m > 1$, the following holds:*

$$\forall \beta \in [d]_0^n, \gamma \in [m]_0^n : \ \delta_\beta(\gamma) \leq \prod_{j=1}^{n}(b_j - a_j)^{\gamma_j} \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^{n} \frac{(\gamma_j - 1)^2}{d}.$$

*Proof.* See Section 3.8. □

Similarly to the univariate case we can relate the Bernstein approximation with values taken by $p$ on a grid.

**Theorem 3.14** ([61, Theorem 2] and [18, Theorem 4])**.** *For $p \in \mathbb{R}[x]_m$ and $d \geq m \geq 1$, we have*

$$|p_{\min,[\underline{a},\underline{b}](d)} - p^{(d)}_{Ber(a,b)}| \leq \Delta(d)$$

*with*

$$\Delta(d) = \vartheta_p \max(1, |a_1|^m, \ldots, |a_n|^m) \sum_{\gamma \in [m]_0^n} \left( \prod_{j=1}^{n} \|b-a\|_\infty^{\sum_{j=1}^n \gamma_j} \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^{n} \frac{(\gamma_j - 1)^2}{d} \right)$$

$$\text{and} \quad \vartheta_p = \max_{\gamma \in [m]_0^n} \sum_{\substack{\alpha \in [m]_0^n \\ \alpha \geq \gamma}} |p_\alpha| \prod_{j=1}^{n} \binom{\alpha_j}{\gamma_j}.$$

*Proof.* See Section 3.8. □

The following theorem describes a bound on the Handelman approach based on a surrounding box of the polytope $K$. Note that such a box can be found by solving $2n$ linear programs over the polytope $K$, minimizing and maximizing in direction of the coordinate axes.

**Theorem 3.15.** *Let $K$ be a polytope contained in a box $[\underline{a}, \underline{b}]$. Assume that $p_{min,[a,b]}$ is attained at a point in $K$. For $p \in \mathbb{R}[x]_m$ and any integer $d \geq mn$, we have $p_{min,K} - p_{Han,G}^{(dn)} \leq p_{min,K} - p_{Ber(a,b)}^{(d)} \leq \Delta(d)$, where for $\|b-a\|_\infty < 1$,*

$$\Delta(d) \leq \vartheta_p \max(1, |a_1|^m, \ldots, |a_n|^m) \|b-a\|_\infty^2 \sum_{\gamma \in [m]_0^n} \left( \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^{n} \frac{(\gamma_j - 1)^2}{d} \right)$$

*and for $\|b-a\|_\infty \geq 1$,*

$$\Delta(d) \leq \vartheta_p \max(1, |a_1|^m, \ldots, |a_n|^m) \|b-a\|_\infty^{mn} \sum_{\gamma \in [m]_0^n} \left( \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^{n} \frac{(\gamma_j - 1)^2}{d} \right).$$

*Proof.* As we have seen in Lemma 4.1, adding redundant inequalities to the problem does not change the value $p_{Han,G}^{(d)}$. Therefore, we can add the inequalities defining the bounding box. The error bound from theorem 3.14 is valid in presence of these constraints.

For $\|b-a\|_\infty < 1$, we get a quadratic term in $\|b-a\|_\infty$, because there has to be at least one $\gamma_i \geq 2$ in each summand, otherwise the term in parentheses vanishes. □

*Remark* 3.16. We conjecture that similarly strong bounds exist for the general case where the minimum of $p$ over the hypercube does not lie in the polytope. For our purpose, namely to show convergence of the branch and bound method described in the next chapter, the result provided here suffices.

In the next corollary, we provide a somewhat coarser version of the bound from Theorem 3.14 on the unit box to make it easier to compare.

**Corollary 3.17.** *For $[\underline{a}, \underline{b}] = [\underline{0}, \underline{1}]$, the bound from Theorem 3.15 is bounded by*

$$p_{min,K} - p^{(dn)}_{Han,G} \leq p_{min,K} - p^{(d)}_{Ber(a,b)} \leq \frac{\|p\|}{d} n(n+1)^2 m^{nm+4}.$$

*Proof.* The result follows by rigorous bounding of $\vartheta_p$.

$$\begin{aligned}
\vartheta_p &= \max_{\gamma \in [m]_0^n} \sum_{\substack{\alpha \in [m]_0^n \\ \alpha \geq \gamma}} |p_\alpha| \prod_{j=1}^{n} \binom{\alpha_j}{\gamma_j} \leq (m+1)^n \max_{\substack{\alpha,\gamma \in [m]_0^n \\ \alpha \geq \gamma}} |p_\alpha| \prod_{j=1}^{n} \binom{\alpha_j}{\gamma_j} \\
&\leq (m+1)^n \max_{\substack{\alpha,\gamma \in [m]_0^n \\ \alpha \geq \gamma}} \|p\| \frac{(\sum_{j=1}^n \alpha_j)!}{\prod_{j=1}^n \alpha_j!} \frac{\prod_{j=1}^n \alpha_j!}{\prod_{j=1}^n (\alpha_j - \gamma_j)! \gamma_j!} \\
&\leq (m+1)^n \|p\| (nm)!
\end{aligned}$$

$$\begin{aligned}
\Delta(d) &\leq \vartheta_p \sum_{\gamma \in [m]_0^n} \left( \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^{n} \frac{(\gamma_j - 1)^2}{d} \right) \\
&\leq \vartheta_p (m+1)^n n \frac{(m-1)^2}{d} \leq \frac{\|p\|}{d} (m+1)^{2n} (nm)^{nm+2}.
\end{aligned}$$

□

Note that the bound in theorem 3.15 depends on the length of the longest edge of a box $[\underline{a}, \underline{b}]$, surrounding the polytope over which we optimize. All other factors in the bound are bounded in size. In other words, we can expect tighter and tighter approximations, as we divide along the dimension of the longest edge in the Handelman procedure. Moreover, with repeated subdivision of the interval, the minimum over the optimal values on the subintervals converges to the minimum of the objective polynomial.

## 3.6 Deriving Degree Bounds from Error Bounds

The error bound in Theorem 3.15 can be used to design a degree bound for the Handelman approach. This is a common practice, the degree bounds in Section 3.1 were developed in the same way.

**Lemma 3.18.** *Let $p \in \mathbb{R}[x]$ be positive on the polytope $P$, which is described by linear polynomials $G = \{g_1, \ldots, g_k\}$. Assume that the minimum of $p$ over a bounding box is attained at a point in $P$. Let $\Delta(d)$ be defined as in Theorem 3.15. Choose $d$ such that $p_{min,P} - \Delta(d)$ is positive. Then $p \in H_d(G)$.*

*Proof.* We have

$$p_{min,P} - p^{(d)}_{Han,G} \leq \Delta(d)$$

and by definition

$$p(x) - p^{(d)}_{Han,G} \in H_d(G).$$

This implies

$$p(x) + (p_{min,P} - p^{(d)}_{Han,G}) - p_{min,P} \in H_d(G),$$

and thus

$$p(x) - (p_{min,P} - \Delta(d)) \in H_d(G).$$

Since $p_{min,P} - \Delta(d)$ is positive by assumption, $p(x) \in H_d(G)$. □

Note that in the special setting where the optimum over a bounding box lies in the polytope $P$, this yields an alternative proof for Handelman's Theorem 2.16, since $\Delta(d)$ converges to zero as $d$ goes to infinity.

## 3.7 Bernstein and Handelman

From the presentation in the last sections, one could get the impression that the Bernstein approximation and the Handelman approximation are the same when optimizing over a box. This observation is indeed close to the truth, we will see in Corollary 3.20 that the Bernstein bound coincides with the Handelman bound if we restrict the Handelman representation not by total degree but by a certain degree in each variable.

In fact, it is due to the special structure of the Handelman representation on the box that we only need polynomials of the highest degree in each variable for the representation.

**Lemma 3.19.** *Let $p \in \mathbb{R}[x]$ and let $P$ be a box as in* (2.9) *described by linear polynomials $g_1, \ldots, g_k$. If $p$ admits a representation*

$$p(x) = \sum_{\alpha \in \mathbb{N}^k} \lambda_\alpha g^\alpha, \lambda_\alpha \geq 0$$

*with $g^\alpha \in \mathbb{R}[x]_{(d,\ldots,d)}$, then it also admits such a representation for which $g^\alpha \in \mathbb{R}[x]_{(d,\ldots,d)} \setminus \mathbb{R}[x]_{(d-1,\ldots,d-1)}$.*

*Proof.* Since the feasibility region is a box, the polynomials $g^\alpha$ above are Bernstein polynomials (up to a positive factor consisting of binomial coefficients). The same reasoning as in the proof of lemma 3.7 applies. □

This has direct implications on the relation of the Bernstein bound and the Handelman bound.

**Corollary 3.20.** *Let $p \in \mathbb{R}[x]$ and let $P$ be a box as in (2.9) described by linear polynomials $g_1, \ldots, g_k$.*

*When restricting to a certain degree $d$ in every variable instead of a certain total degree, the Bernstein bound and the Handelman bound coincide, that is, the optimal value $t^*$ of the optimization problem*

$$\begin{aligned} \text{maximize} \quad & t \\ s.t. \quad & p(x) - t = \sum_{\alpha \in \mathbb{N}^m} \lambda_\alpha g^\alpha, \\ & \lambda_\alpha \geq 0, \\ & g^\alpha \in \mathbb{R}[x]_{(d,\ldots,d)} \end{aligned}$$

*is equal to $p^{(d)}_{Ber,P}$.*

*Proof.* By lemma 3.19, the same optimal value $t^*$ can be achieved with polynomials $g^\alpha \in \mathbb{R}[x]_{(d,\ldots,d)} \setminus \mathbb{R}[x]_{(d-1,\ldots,d-1)}$, which are positive multiples of the Bernstein polynomials of degree $d$. Therefore,

$$p(x) - t^* = \sum_{\beta \in [d]_0^n} \lambda'_\beta \; P_{\beta,d}(x, a, b)$$

with non-negative coefficients $\lambda'_\beta$. Using that the Bernstein polynomials form a partition of unity (Lemma 3.6), we get

$$\begin{aligned} p(x) &= \sum_{\beta \in [d]_0^n} \lambda'_\beta \; P_{\beta,d}(x, a, b) + t^* \sum_{\beta \in [d]_0^n} P_{\beta,d}(x, a, b) \\ &= \sum_{\beta \in [d]_0^n} (\lambda'_\beta + t^*) \; P_{\beta,d}(x, a, b) \end{aligned}$$

and thus $p^{(d)}_{Ber,P} = min_{\beta \in [d]_0^n} \lambda'_\beta + t^* \geq t^*$.

The reverse inequality is shown by the argument in Section 3.3. □

This is remarkable, since no optimization techniques have to be used to compute the Bernstein bound. There exist efficient methods to find the Bernstein bound in practice. In branch and bound methods like the one we will describe in the next chapter, difference schemes can be used to successively find the Bernstein bound on partition sets.

But this does not mean that the Bernstein approach is always superior to the Handelman approach. The Handelman Positivstellensatz admits convex polytopes as feasible regions and can thus be used in a much more general setting.

## 3.8 Proofs

In this section we collect the proofs omitted earlier in this chapter.

*Proof (of Theorem 3.10).* Since $B_d(1,a,b) = 1$ and $B_d(x,a,b) = x$ (see Lemma 3.6), $\delta_j(0) = \delta_j(1) = 0$. This means that we can restrict our considerations to the case $s \geq 2$. Using Lemma 3.8, we have

$$\begin{aligned}
&B_d((x-a)^s,a,b) - (x-a)^s \\
=&\sum_{j=0}^{d} \left[\frac{j}{d}(b-a)\right]^s p_{j,d}(x,a,b) - \sum_{j=s}^{d} \frac{\binom{j}{s}}{\binom{d}{s}}(b-a)^s p_{j,d}(x,a,b) \\
=&\sum_{j=0}^{s-1} \left[\frac{j}{d}(b-a)\right]^s p_{j,d}(x,a,b) + \sum_{j=s}^{d} \left(\left[\frac{j}{d}(b-a)\right]^s - \frac{\binom{j}{s}}{\binom{d}{s}}(b-a)^s\right) p_{j,d}(x,a,b).
\end{aligned}$$

By comparing coefficients, for $0 \leq j < s$ we get

$$\delta_j(s) = \left[\frac{j}{d}(b-a)\right]^s \leq \frac{(s-1)^2}{d^2}(b-a)^s \leq \frac{(s-1)^2}{d}(b-a)^s.$$

and for $2 \leq s \leq j$ we get

$$\begin{aligned}
\delta_j(s) &= \left(\left[\frac{j}{d}(b-a)\right]^s - \frac{\binom{j}{s}}{\binom{d}{s}}(b-a)^s\right) \\
&= \left(\left[\frac{j}{d}\right]^s - \frac{j!(d-s)!}{(j-s)!d!}\right)(b-a)^s \\
&= \left(\frac{j}{d}\right)^s \left(1 - \frac{\left(1-\frac{1}{j}\right)\cdots\left(1-\frac{s-1}{j}\right)}{\left(1-\frac{1}{d}\right)\cdots\left(1-\frac{s-1}{d}\right)}\right)(b-a)^s \\
&\leq \left(\frac{j}{d}\right)^s \left(1 - \left(1-\frac{1}{j}\right)\cdots\left(1-\frac{s-1}{j}\right)\right)(b-a)^s \\
&\leq \left(\frac{j}{d}\right)^s \left(1 - \left(1-\frac{s-1}{j}\right)^{s-1}\right)(b-a)^s.
\end{aligned}$$

Now apply the mean value theorem to $f(x) := (1-x)^{s-1}$. We have $f'(x) = -(s-1)(1-x)^{s-2}$. Choose $a = 0, b = \frac{s-1}{j}$.

Then

$$\begin{aligned}
&\frac{f(b)-f(a)}{b-a} = f'(x_0) \quad \text{for some } x_0 \in [a,b] \\
\Leftrightarrow \quad &\frac{\left(1-\frac{s-1}{j}\right)^{s-1} - 1}{\frac{s-1}{j} - 0} = -(s-1)(1-x_0)^{s-2} \\
\Leftrightarrow \quad &1 - (1 - \frac{s-1}{j})^{s-1} = \frac{(s-1)^2}{j}(1-x_0)^{s-2} \\
&\leq \frac{(s-1)^2}{j}
\end{aligned}$$

and it follows that

$$\begin{aligned}\delta_j(s) &\leq \left(\frac{j}{d}\right)^{s-1} \frac{(s-1)^2}{j}(b-a)^s \\ &\leq \frac{(s-1)^2}{d}(b-a)^s.\end{aligned}$$

□

*Proof (of Theorem 3.11).* By definition of $p^{(d)}_{Ber(a,b)}$, $\Delta(d)$ can be bounded by the corresponding $j$-th differences

$$\Delta(d) \leq \max_{j\in[m]_0} \Delta_j(d),$$
$$\text{where } \forall j \in [m]_0\ \Delta_j(d) := p\left(a + \frac{j}{d}(b-a)\right) - b_j^{(d)}.$$

Again we use the Bernstein basis to express the difference between the Bernstein approximation of the polynomial $p$ and $p$ itself

$$\begin{aligned}B_d(p,a,b) - p(x) &= \sum_{j=0}^{d} \left[p\left(a + \frac{j}{d}(b-a)\right) - b_j^{(d)}\right] p_{j,d}(x,a,b) \\ &= \sum_{j=0}^{d} \Delta_j(d)\ p_{j,d}(x,a,b).\end{aligned}$$

Using equation (3.7), we get

$$B_d(p,a,b) - p(x) = \sum_{j=0}^{d}\sum_{s=0}^{m} c_s\delta_j(s)\ p_{j,d}(x,a,b).$$

Therefore, we see that

$$\Delta_j(d) = \sum_{s=0}^{m} c_s\delta_j(s).$$

Using the definition of the $c_s$ in (3.4), the bound on $\delta_j(s)$ from Theorem 3.10 and the fact that $\delta_0(s) = \delta_1(s) = 0$, we get [1]

$$\begin{aligned}\Delta_j(d) &\leq \sum_{s=2}^{m}\left(\sum_{j=s}^{m}|p_j|\binom{j}{s}a^{j-s}\right)(s-1)^2\frac{(b-a)^s}{d} \\ &\leq \vartheta_p \max(|a|^m, 1)\sum_{s=2}^{m}(s-1)^2\frac{(b-a)^s}{d}.\end{aligned}$$

□

[1] The factor $\vartheta_p$ comes from the representation of the polynomial $p$ in terms of the basis $(x-a)^s$ as explained in equation (3.4). In [78], the factor is defined as $\sum_{s=2}^{n}|p_s|$, while [18, 61] contain the version shown here. We believe the additional binomial coefficients cannot be omitted.

*Proof (of Theorem 3.13).*

$$
\begin{aligned}
&B_d(\prod_{j=1}^n (x_j-a_j)^{\gamma_j}, a, b) - \prod_{j=1}^n (x_j-a_j)^{\gamma_j} \\
&= \left[ \sum_{\beta\in[d]_0^n} \prod_{j=1}^n \left(\frac{\beta_j}{d}\right)^{\gamma_j} (b_j-a_j)^{\gamma_j} - \sum_{\substack{\beta\in[d]_0^n \\ \beta\geq\gamma}} \prod_{j=1}^n \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} (b_j-a_j)^{\gamma_j} \right] P_{\beta,d}(x,a,b) \\
&= \left[ \sum_{\beta\in[d]_0^n} \prod_{j=1}^n \left(\frac{\beta_j}{d}\right)^{\gamma_j} - \sum_{\substack{\beta\in[d]_0^n \\ \beta\geq\gamma}} \prod_{j=1}^n \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} \right] \prod_{j=1}^n (b_j-a_j)^{\gamma_j} P_{\beta,d}(x,a,b).
\end{aligned}
$$

If $\beta_i < \gamma_i$ for some $i \in [n]$, and thus $\gamma_i \geq 1$, we have

$$
\begin{aligned}
\delta_\beta(\gamma) &= \prod_{j=1}^n \left(\frac{\beta_j}{d}\right)^{\gamma_j} (b_j-a_j)^{\gamma_j} \\
&\leq \left(\frac{\gamma_i-1}{d}\right)^2 \prod_{j=1}^n (b_j-a_j)^{\gamma_j} \\
&\leq \sum_{j=1}^n (\gamma_j-1)^2 \frac{1}{d} \prod_{j=1}^n (b_j-a_j)^{\gamma_j},
\end{aligned}
$$

if $\beta \geq \gamma$ we have

$$
\delta_\beta(\gamma) = \left[ \prod_{j=1}^n \left(\frac{\beta_j}{d}\right)^{\gamma_j} - \prod_{j=1}^n \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} \right] \prod_{j=1}^n (b_j-a_j)^{\gamma_j}.
$$

Since

$$
\prod_{j=1}^n \left(\frac{\beta_j}{d}\right)^{\gamma_j} - \prod_{j=1}^n \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} \leq \prod_{\substack{j=1 \\ \gamma_j\notin\{0,1\}}}^n \left(\frac{\beta_j}{d}\right)^{\gamma_j} - \prod_{\substack{j=1 \\ \gamma_j\notin\{0,1\}}}^n \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}},
$$

we can bound $\delta_\beta(\gamma)$ by

$$
\delta_\beta(\gamma) \leq \left[ \prod_{\substack{j=1 \\ \gamma_j\notin\{0,1\}}}^n \left(\frac{\beta_j}{d}\right)^{\gamma_j} - \prod_{\substack{j=1 \\ \gamma_j\notin\{0,1\}}}^n \frac{\binom{\beta_j}{\gamma_j}}{\binom{d}{\gamma_j}} \right] \prod_{j=1}^n (b_j-a_j)^{\gamma_j}.
$$

Similarly to the univariate case, the bound can be further estimated by

$$
\delta_\beta(\gamma) \leq \prod_{\substack{j=1 \\ \gamma_j\notin\{0,1\}}}^n \left[ \left(\frac{\beta_j}{d}\right)^{\gamma_j} \left( 1 - \prod_{\substack{j=1 \\ \gamma_j\notin\{0,1\}}}^n \left(1-\frac{\gamma_j-1}{\beta_j}\right)^{\gamma_j} \right) \right] \prod_{j=1}^n (b_j-a_j)^{\gamma_j}.
$$

Using a generalized Bernoulli inequality[2] we get

$$\begin{aligned}
\delta_\beta(\gamma) &\le \left[\prod_{\substack{j=1\\ \gamma_j\notin\{0,1\}}}^{n} \left(\frac{\beta_j}{d}\right)^{\gamma_j} \left(1-\left(1-\sum_{\substack{i=1\\ \gamma_i\notin\{0,1\}}}^{n} \frac{(\gamma_i-1)^2}{\beta_i}\right)\right)\right] \prod_{j=1}^{n}(b_j-a_j)^{\gamma_j} \\
&= \left[\prod_{\substack{j=1\\ \gamma_j\notin\{0,1\}}}^{n} \left(\frac{\beta_j}{d}\right)^{\gamma_j} \left(\sum_{\substack{i=1\\ \gamma_i\notin\{0,1\}}}^{n} \frac{(\gamma_i-1)^2}{\beta_i}\right)\right] \prod_{j=1}^{n}(b_j-a_j)^{\gamma_j} \\
&= \left[\sum_{\substack{i=1\\ \gamma_i\notin\{0,1\}}}^{n} \left(\left(\prod_{\substack{j=1\\ \gamma_j\notin\{0,1\}}}^{n} \left(\frac{\beta_j}{d}\right)^{\gamma_j}\right)\left(\frac{\beta_i}{d}\right)^{\gamma_i-1} \frac{(\gamma_i-1)^2}{d}\right)\right] \prod_{j=1}^{n}(b_j-a_j)^{\gamma_j} \\
&\le \sum_{\substack{i=1\\ \gamma_i\notin\{0,1\}}}^{n} (\gamma_i-1)^2 \,\frac{1}{d}\, \prod_{j=1}^{n}(b_j-a_j)^{\gamma_j}.
\end{aligned}$$

□

*Proof of Theorem 3.14.* Again, $\Delta(d) \le \max_{\beta\in[d]_0^n} \Delta_\beta(d)$, where

$$\Delta_\beta(d) = p\left(a_1+\frac{\beta_1}{d}(b_1-a_1),\ldots,a_n+\frac{\beta_n}{d}(b_n-a_n)\right) - b_\beta^{(d)}.$$

We have

$$\begin{aligned}
B_d(p,a,b)-p(x) &= \sum_{\beta\in[d]_0^n} \Delta_\beta(d) P_{\beta,d}(x,a,b) \\
&= \sum_{\beta\in[d]_0^n}\sum_{\gamma\in[m]_0^n} c_\gamma \delta_\beta(\gamma) P_{\beta,d}(x,a,b),
\end{aligned}$$

and therefore

$$\begin{aligned}
\Delta_\beta(d) &= \sum_{\gamma\in[m]_0^n} c_\gamma \delta_\beta(\gamma) \\
&\le \sum_{\gamma\in[m]_0^n} \left(\sum_{\substack{\alpha\in[m]_0^n\\ \alpha\ge\gamma}} |p_\alpha| \prod_{j=1}^{n} \binom{\alpha_j}{\gamma_j} a_j^{\alpha_j-\gamma_j}\right) \left(\prod_{j=1}^{n}(b_j-a_j)^{\gamma_j} \sum_{\substack{j=1\\ \gamma_j\notin\{0,1\}}}^{n} \frac{(\gamma_j-1)^2}{d}\right)
\end{aligned}$$

[2] [27, p. 60] Generalized Bernoulli inequality: If $\alpha,\beta,\ldots,\delta$ are greater than $-1$ and all positive or all negative, then $(1+\alpha)(1+\beta)\cdots(1+\delta) > 1+\alpha+\beta+\ldots+\delta$

$$\leq \vartheta_p \max(1, |a_1|^m, \dots, |a_n|^m) \sum_{\gamma \in [m]_0^n} \left( \prod_{j=1}^n (b_j - a_j)^{\gamma_j} \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^n \frac{(\gamma_j - 1)^2}{d} \right)$$

$$\leq \vartheta_p \max(1, |a_1|^m, \dots, |a_n|^m) \sum_{\gamma \in [m]_0^n} \left( \prod_{j=1}^n \|b - a\|_\infty^{\sum_{j=1}^n \gamma_j} \sum_{\substack{j=1 \\ \gamma_j \notin \{0,1\}}}^n \frac{(\gamma_j - 1)^2}{d} \right).$$

□

# 4 Improving Positivstellensatz Relaxations by Adding Inequalities and Branching

In Chapter 2, we described how Positivstellensätze can be used to construct relaxations for polynomial optimization problems. With higher degree bounds, the complexity of these relaxations also increases. Therefore, one tries to improve the relaxations without changing the degree bound. This is usually done by adding additional polynomials to the description of the set $K$ over which the objective function is to be optimized.

In Section 4.1, we discuss the impact of adding redundant polynomials to the description of the feasible set. Since our main focus in this work lies on the Handelman relaxation, we are primarily interested in adding linear inequalities.

Instead of adding redundant linear polynomials, one can split the problem into two parts by adding a non-redundant linear constraint. As we have seen in the last chapter, we can expect the bounds to improve when the feasible set becomes smaller. We make use of this fact and introduce a new branch and bound method based on the Handelman relaxation in Section 4.2.1. We prove that the suggested approach converges. Further variants of the approach (an additional splitting rule) are discussed in Section 4.3.

Our main interest throughout is the Handelman relaxation, therefore we will mainly be concerned with polynomial optimization over a polytope

$$\begin{aligned} \text{minimize} \quad & p(x) \\ \text{s.t.} \quad & x \in P, \\ & P = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0,\ g_i \text{ linear}\}. \end{aligned} \tag{4.1}$$

## 4.1 Adding Redundant Polynomials

In applications one tries to modify the polynomial optimization problem that is to be solved, in order to reduce the error introduced by the relaxation. The most common approach is to enrich the constraint set by adding redundant polynomials to the description of the set $K$. This means that the original polynomial optimization problem remains unchanged. At first sight, it is not clear why this should improve the relaxation.

In [52], Lasserre provides a rationale for doing so. He relates the Positivstellensatz relaxation to the dual method of multipliers in nonlinear programming. The

resulting Lagrangian relaxation of the enriched polynomial optimization problem differs from the Lagrangian relaxation of the original problem. Basically, additional polynomials in the description of the feasibility region lead to a higher degree of freedom in the Positivstellensatz representation of the objective polynomial.

### 4.1.1 Redundant Polynomials in the Linear Setting

We begin by exploring the impact of adding linear redundant polynomials. The following two lemmata show that in the linear setting of problem (4.1) linear polynomials do not have an impact on the quality of the relaxation. In view of the dual method described above, this does not come as a surprise, linearly dependent constraints should not increase the degree of freedom. The situation changes when the feasible region is not described by linear inequalities.

**Lemma 4.1.** *Let $P$ be a non-empty $\mathcal{H}$-polytope described by linear polynomials $G = \{g_1, \ldots, g_k\}$. The Handelman approximation approach* (2.24) *is invariant under adding redundant linear constraints to the description $G$ of $P$.*

*Proof.* Let $g_{k+1}$ be a redundant linear polynomial for the description of $P$, that means, $g_{k+1}(x) \geq 0$ for all $x \in P$. We show that $H_d(G) = H_d(G \cup \{g_{k+1}\})$, which proofs the statement.

By definition, we have $H_d(G) \subseteq H_d(G \cup \{g_{k+1}\})$. For the reverse inclusion, we have to show that terms of the form $g_1^{\alpha_1} \cdots g_k^{\alpha_k} g_{k+1}^{\alpha_{k+1}}$ with $\sum_{i=1}^{k+1} \alpha_i \leq d$ are contained in $H_d(G)$. It follows from Farkas' Lemma 2.1, that $g_{k+1}$ is a nonnegative combination of the polynomials in $G$ :

$$g_{k+1} = \lambda_0 + \sum_{i=1}^{k} \lambda_i g_i, \quad \lambda_0 \geq 0, \ldots, \lambda_k \geq 0.$$

This implies that $g_{k+1}^{\alpha_{k+q}} \subset H_{\alpha_{k+1}}(G)$. Using Lemma 2.19, it immediately follows that $g_1^{\alpha_1} \cdots g_k^{\alpha_k} g_{k+1}^{\alpha_{k+1}} \in H_d(G)$ and thus $H_d(G \cup \{g_{k+1}\}) \subseteq H_d(G)$. □

The proof of Lemma 4.1 also implies the following: Any redundant linear inequalities can be eliminated from the description of $P$ without changing the value of the optimization problem (2.24). Reduction of the number of inequalities also reduces the complexity of the optimization problem.

A similar statement can be made about the Putinar approach when restricting to linear constraints.

**Lemma 4.2.** *Let $P$ be a non-empty $\mathcal{H}$-polytope described by linear polynomials $G = \{g_1, \ldots, g_k\}$. The Putinar approximation approach* (2.22) *is invariant under adding redundant linear constraints to the description $G$ of $P$, that is $p^d_{Put}(G) = p^d_{Put}(G \cup g_{k+1})$, where $g_{k+1}$ is a redundant linear constraint.*

*Proof.* Using the representation of a redundant inequality from the Farkas' Lemma 2.1, $g_{k+1} = \lambda_0 + \sum_{i=1}^{k} \lambda_i g_i, \quad \lambda_0 \geq 0, \ldots, \lambda_k \geq 0$, we see that every Putinar representation in terms of the constraint set $G \cup \{g_{k+q}\}$ can be transformed into a Putinar representation in terms of $G$. Given $\sigma_0 + \sum_{i=1}^{k+1} \sigma_i g_i$ with $\sigma_i \in \Sigma, deg(\sigma_i g_i) \leq d$, we insert the Farkas representation to get

$$\begin{aligned}
&\sigma_0 + \sum_{i=1}^{k} \sigma_i g_i + \sigma_{k+1}(\lambda_0 + \sum_{i=1}^{k} \lambda_i g_i) \\
=&(\sigma_0 + \sigma_{k+1}\lambda_0) + \sum_{i=1}^{k} (\sigma_i + \sigma_{k+1}\lambda_i) g_i \\
=&\sigma_0' + \sum_{i=1}^{k} \sigma_i' g_i.
\end{aligned}$$

Since the new multipliers are a positive combination of the former, they are still sums of squares. The degree bounds $deg(\sigma_i' g_i) \leq d$ are still satisfied since all polynomials $g_1, \ldots, g_{k+1}$ are linear. □

From the proofs of the last two statements it is obvious that the statements also hold if the optimization is not over a polytope, but the redundant linear inequalities can be expressed by other linear inequalities.

However, it is not generally true that linear redundant inequalities do not have an effect. For nonlinear constraints, adding linear redundant inequalities can improve the relaxation. The next example shows this behavior for the Putinar approach.

**Example 4.3.** Consider the minimization of the polynomial $p(x, y) = x$ over the Bean curve described by the constraint

$$G = \left\{ \left(x + \frac{1}{2}\right) \left( \left(x + \frac{1}{2}\right)^2 + y^2 \right) - \left(x + \frac{1}{2}\right)^4 - \left(x + \frac{1}{2}\right)^2 y^2 - y^4 \right\}.$$

The constraint polynomial describes a curve with a singularity at $(-0.5, 0)$. This leads to an inaccuracy in the relaxation. We have

$$\begin{aligned}
p_{Put}^{(4)}(G) &= -0.6315, \\
p_{Put}^{(6)}(G) &= -0.5498, \\
p_{Put}^{(8)}(G) &= -0.5289.
\end{aligned}$$

The true optimal value is reached when adding the redundant constraint $x - \frac{1}{2} \geq 0$ to the constraint set:

$$p_{Put}^{(4)}(G \cup \{x - 1/2\}) = -0.5000.$$

The varieties of the constraints are depicted in Figure 4.1.

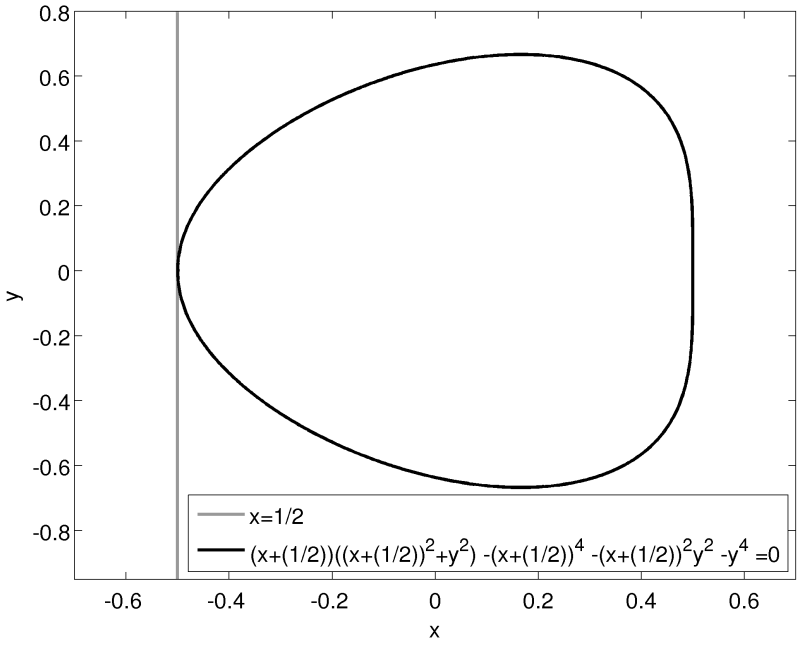


Figure 4.1: Bean Curve

While the example might seem constructed on the first view, it is commonly known that the Putinar approach does not lead to exact results around singularities of the variety described by the constraints (see [22, Theorem 4.3] ). If such singular points are known a priori, it is a good idea to enhance the description of the set $K$ with supporting linear inequalities.

### 4.1.2 Redundant Polynomials in the Non-Linear Setting

For this section we leave the restricted setting of polytopal feasible sets. While the convergence results for Handelman's approach only apply in the polytopal setting, adding redundant non-linear equations is of course not forbidden. In this more general setting, the idea of adding redundant polynomials to the description of the set $K$ when applying Putinar's approach has been pursued in several works.

The most prominent improvement is to add a polynomial of the form $N - \sum_{i=1}^{n} x_i^2$ with large enough $N$ in order to guarantee that the quadratic module generated by the constraints is Archimedean. This may be necessary for cases in which the set $K$ is compact, but the preconditions for Putinar's Positivstellensatz are not satisfied. The idea of adding this constraint was inspired by Putinar's theoretical result. In concrete applications, it has also turned out to improve the speed of convergence. This approach has been described by Lasserre in [49, 50] and by Lasserre and Henrion in [30].

| $p^{(d)}_{Han,G}$ | $p^{(d)}_{Han,G'}$ | $p(x,y)$ |
|---|---|---|
| $-0.3125$ | $-0.1708$ | $-0.72\,x^3y - 0.37\,xy^3 + 0.72\,y^2x^2 - 0.13\,x^2y - 0.88\,xy^2$ |
| $-0.1317$ | $-0.0829$ | $0.77\,x^3y - 0.49\,xy^3 - 0.42\,y^2x^2 - 0.55\,x^2y + 0.18\,xy^2$ |
| $-0.1317$ | $-0.0782$ | $0.13\,x^3y + 0.73\,xy^3 - 0.56\,y^2x^2 - 0.40\,x^2y + 0.17\,xy^2$ |
| $-0.0600$ | $-0.0300$ | $0.25\,x^3y - 0.08\,xy^3 + 0.30\,y^2x^2 + 0.96\,x^2y - 0.16\,xy^2$ |
| $-0.0200$ | $-0.0120$ | $0.95\,x^3y + 0.04\,xy^3 + 0.09\,y^2x^2 - 0.24\,x^2y + 0.70\,xy^2$ |
| $-0.2516$ | $-0.1688$ | $0.09\,x^3y + 0.63\,xy^3 - 0.03\,y^2x^2 - 0.77\,x^2y - 0.71\,xy^2$ |
| $-0.1525$ | $-0.1171$ | $0.57\,x^3y - 0.46\,xy^3 + 0.11\,y^2x^2 - 0.73\,x^2y - 0.15\,xy^2$ |
| $-0.0850$ | $-0.0425$ | $-0.88\,x^3y + 0.59\,xy^3 + 0.12\,y^2x^2 + 0.54\,x^2y + 0.99\,xy^2$ |
| $-0.4000$ | $-0.2000$ | $-0.82\,x^3y + 0.44\,xy^3 + 0.99\,y^2x^2 - 0.78\,x^2y - 0.09\,xy^2$ |

Table 4.1: Computational test with added redundant inequality.

Another common approach is to add optimality conditions known from the theory of nonlinear programming. In [68], the well known KKT-conditions were used to enforce finite convergence of the Putinar approach. This result, however, comes at the cost of adding additional variables to model these conditions.

In [66], Nie proposes a new type of relaxation, also using the KKT-conditions, but without additional variables. The disadvantage of this approach is that the number of constraints grows rapidly. More recently these approaches have been discussed in [1].

As we have seen, adding redundant linear polynomials to the description of the polytope does not have any effect. But adding redundant polynomials of higher degree may very well make a difference, as the following example shows.

**Example 4.4** (cp. [51], Ex. 5.5)**.** Let $P = \{x \in \mathbb{R} :\ 0 \le x \le 1\}$ and the objective function be $f : x \mapsto x(x-1)$. The optimal value of the minimization problem is $-\frac{1}{4}$, attained at $x^* = \frac{1}{2}$.

For $G = \{x, 1-x\}$, the Handelman-hierarchy (2.24) of degree $d = 2$ overshoots and returns an optimal value of $-\frac{1}{2}$.

The objective can be improved by enriching the constraint set G with the quadratic polynomial $q(x) = x^2 - x + \frac{1}{4} = (x - \frac{1}{2})^2$. The polynomial $q$ is redundant: since it is a square, it is positive on $\mathbb{R}$. The Handelman-hierarchy enhanced with $q$ yields the true optimal value already for $d = 2$.

It is remarkable that the linear-factor of $q$ is a non-redundant polynomial in the description of $P$. Only by squaring, it becomes redundant.

To make the impact of such equations more tangible, we performed a computational experiment on randomly generated polynomials. We optimized the polynomials over the simplex; see Table 4.1. Here $G$ denotes the constraint set $G = \{x, y, 1-x-y\}$, while $G'$ is the enriched constraint set $G' = G \cup \{(x-y)^2\}$.

We did not see any difference for linear and quadratic polynomials. The degree four polynomials in Table 4.1 however improve significantly. This is due to the special structure of the polynomials above.

In the same spirit, Ghaddar, Vera and Anjos [19] describe a method to dynamically generate additional redundant inequalities in Putinar's relaxation. Without raising the relaxation order, there method enriches the constraint set. They proof convergence of this method for some important special cases such as binary polynomial programs with quadratic objective function and linear constraints.

## 4.2 Subdivision Algorithm

The use of branch and bound algorithms has a long tradition in the field of optimization. Following the principal of divide and conquer, the first easy to implement algorithm was introduced by Land and Doig [44] in 1960. The algorithm, as well as several refinements (see e.g. [11]), aimed mostly at solving integer or mixed integer linear programs. Soon, other applications of branch and bound algorithms were found. Lawler and Wood describe applications to nonlinear programming and the traveling salesman problem in their 1966 survey [54].

Branch and bound like subdivision algorithms also play an important role in solving polynomial optimization problems. A prominent example is the Bernstein algorithm tailored to solve polynomial optimization problems over intervals [45], and over rectangular boxes ([79, 65]). More recent works deal with more general polynomial programming problems but still rely on branching on some hyperrectangular region, see, e.g., [15].

One of the main advantages of branch and bound methods is the generation of lower and upper bounds on the optimum. This allows to get a certified error bound for the result. Such bounds are often very important in practical applications.

In this section we extend the ideas from the Bernstein algorithm. Instead of box constraints, we allow arbitrary polytopes as feasible sets. This is achieved by combining the conventional branch and bound algorithm with Handelman's Positivstellensatz.

### 4.2.1 Branch and Bound Algorithms

As before, the method of choice to find a solution of problem (4.1) will be Handelman's Positivstellensatz. Instead of elevating the degree to get better approximations, we incorporate a low degree relaxation into a branch and bound algorithm. The beautiful structure of these relaxations allow to obtain convergence results for the branch and bound scheme.

The basic idea in branch and bound methods is to subdivide the whole search space in an optimization problem into smaller and smaller subsets and to calculate a bound on each subset. Then subsets on which the bound is higher (for minimization problems) than a known solution are excluded from the search space. The search space is further subdivided until the solution lies in a specified tolerance.

Our presentation is loosely based on the classical textbook by Tuy and Horst [36] and the more recent survey on geometric branch and bound methods by Scholz [82].

Algorithm 1 is a prototype branch and bound method in pseudocode. We will now describe the components of the method in detail.

The essential underlying idea of a branch and bound algorithm is the subdivision of the feasible region into a partition of sets, which we define now.

**Definition 4.5** ([36, Def. IV.1])**.** Let $H$ be a subset of $\mathbb{R}^n$ and $I$ be a finite set of indices. A set

$$\{H_i \mid i \in I\}$$

of subsets of $H$ is said to be a partition of $H$ if

$$H = \bigcup_{i \in I} H_i,\ H_i \cap H_j = \partial H_i \cap \partial H_j\ \forall i, j \in I, i \neq j,$$

where $\partial H_i$ denotes the (relative) boundary of $H_i$.

We will mainly focus on the case where the partition sets are hyperrectangular sets (boxes) as defined in (2.9). The use of hyperrectangular sets fits well with the definition that we used for the Bernstein basis. As discussed in remark 3.9, Bernstein polynomials can also be defined over simplices.

When allowing arbitrary polytopes $P$ as feasible sets for the optimization problem (4.1), it may happen that the hyperrectangular sets do not intersect with $P$ or intersect only partially. For the further discussion, it is useful to define the following.

**Definition 4.6** ([36, Def. IV.2])**.** Let $P$ be the feasibility region of the optimization problem. A partition set $H$ satisfying $H \cap P = \emptyset$ is called infeasible; a partition set $H$ satisfying $H \cap P \neq \emptyset$ is called feasible.

Next to the partitioning, the most important component of a branch and bound method is the bounding method. This is the method that generates lower bounds for the objective function on a given partition set $H$. In the prototype algorithm 1, we denote the lower bound on $H$ by $p_{lb,H}$.

Additionally, an upper bound denoted by $p_{ub,H}$ on the objective is computed. By comparing the upper and the lower bound, the quality of the bound can be assessed. The upper bound can be generated by evaluating the function on a point or it can be calculated from known error bounds on the lower bound.

A common bounding method for polynomial optimization problems over boxes is the Bernstein method. We discussed how to generate a lower bound for a polynomial optimization problem over a box in the last chapter. The error bound for the Bernstein approach (Theorem 3.15) allows to calculate an upper bound on the minimum, since we have that

$$p^{(d)}_{Ber,[\underline{a},\underline{b}]} \leq p_{min,[\underline{a},\underline{b}]} \leq p^{(d)}_{Ber,[\underline{a},\underline{b}]} \cdot \Delta(d).$$

Alternatively, evaluating the objective polynomial on a feasible point inside the box also gives an upper bound.

**Algorithm 1** Prototype Branch and Bound Algorithm

**Initialization**
$\mathcal{K}_0 \leftarrow \{P\}$
Calculate $p_{lb,P}(0) = p_{lb,K}$
Calculate $p_{ub,P}(0) = p_{ub,K}$
$j \leftarrow 0$
**while** $\mathcal{K}_j \neq \emptyset$ **do**
$j \leftarrow j + 1$
**Selection**
Select subset $\mathcal{H}_j$ of elements from $\mathcal{K}_{j-1}$ to partition
**Subdivision**
Split each element in $\mathcal{H}_j$ according to partition rule
Let $\mathcal{H}'_j$ denote the set of partitioned elements
**Deletion by Infeasibility**
Delete each $H \in \mathcal{H}'_j$ from $\mathcal{H}'_j$ for which $H \cap P = \emptyset$
$\mathcal{K}_j \leftarrow \mathcal{K}_{j-1} \setminus \mathcal{H}_j \cup \mathcal{H}'_j$
**for** each $H$ in $\mathcal{H}'_j$ **do**
**Bounding**
Calculate $p_{lb,H}$
Calculate $p_{ub,H}$
**end for**
$p_{ub,P}(j) \leftarrow \min(p_{ub,H} \mid H \in \mathcal{K}_j)$,
$p_{lb,P}(j) \leftarrow \min(p_{lb,H} \mid H \in \mathcal{K}_j)$
**if** $p_{ub,P}(j) = p_{lb,P}(j)$ **then**
STOP
**end if**
**Prune**
**for** all $H \in \mathcal{K}_j$ **do**
**if** $p_{ub,K} \leq p_{lb,P}(j)$ **then**
$\mathcal{K}_j = \mathcal{K}_j \setminus H$
**end if**
**end for**
**end while**

For the bounding, naturally the choice of the bounding method is a trade-off between accuracy and efficiency. Positivstellensatz relaxations belong to the best global methods known. We overcome the drawback of high complexity in high relaxation orders by restricting the relaxation order (usually to the initial relaxation order). Incorporated into a branch and bound scheme, the initial relaxation order still yields sufficiently good bounds.

For the study of the algorithm, we stick to the most basic rules for selecting partition sets and for subdivision. We propose the following Handelman branch and bound algorithm, refining the corresponding steps in Algorithm 1.

**Handelman branch and bound**

**Selection:** Select an element $H_j$ with the greatest longest edge from $\mathcal{K}_{j-1}$ for partition.

**Subdivision:** Subdivide $H_j$ by bisecting along the longest edge into $H_{j_1}$ and $H_{j_2}$.

**Deletion by Infeasibility:** Use the linear program (2.8) to decide if $H_{j_i} \cap P = \emptyset$ and if nonempty, generate a point $x$ in $H_{j_i} \cap P$.

**Bounding:** Calculate the lower bound using the Handelman relaxation of order mn. Calculate the bound on $H_{j_i} \cap P$ instead of $H_{j_i}$ to get better bounds. Calculate the upper bound by evaluating on the feasible point $x$ returned by the linear program from the Deletion by Infeasibility step.

A branch and bound algorithm is said to be convergent, if

$$p_{lb,P}(j) - p_{ub,P}(j) \xrightarrow[j\to\infty]{} 0.$$

The main goal in this section is to show convergence of the above proposed Handelman branch and bound method.

**Theorem 4.7.** *The Handelman branch and bound method described above converges.*

As for the error bounds discussed in the last chapter, the path to the proof is along the corresponding version based on Bernstein polynomials.

**Bernstein branch and bound**

**Selection, Subdivision and Deletion by Infeasibility:** Similar as above in the Handelman branch and bound method.

**Bounding:** Calculate the lower bound using the Bernstein relaxation of order m. Calculate the upper bound by applying the error bounds as described above for the Bernstein approach.

**Lemma 4.8.** *The Bernstein branch and bound method described above converges.*

To proof these statements, let us review the essentials of the convergence theory developed in [35, 36]. While explaining the theory we will directly discuss the implications for the proposed methods while going along.

Convergence properties are inherently connected with the behaviour of the partition sets. Consider nested (decreasing) sequences of successively refined partition elements, i.e., sequences $\{H_{j_q}\}$ such that

$$H_{j_q} \in \mathcal{H}_{j_q}, \quad H_{j_{q+1}} \subseteq H_{j_q}.$$

Since we deal with general polynomial objective functions without further restrictions (such as, e.g., concavity), we should not expect that the optimal solution can be found in finitely many steps. Thus, infinite decreasing sequences of sets are considered. Note, however, that an $\varepsilon$-approximation (that is $p_{ub,P}(j) - p_{lp,P}(j) \leq \varepsilon$) can sometimes be found much earlier.

Denote by $\delta(H)$ the diameter of $H$, i.e.,

$$\delta(H) := \max(\|x - y\|_2 \mid x, y \in H).$$

Then we can define the following partition property for decreasing sequences.

**Definition 4.9** ([36, Def. IV.10])**.** A subdivision is called exhaustive if

$$\delta(H_{j_q}) \xrightarrow[q\to\infty]{} 0$$

for all decreasing subsequences $\{H_{j_q}\}$ of partition elements generated by the subdivision.

**Lemma 4.10** (Compare [36, Prop. IV.2] for a similar statement concerning simplices)**.** *Let $\{H_{j_q}\}$ be any decreasing sequence of $n$-hyperrectangles generated by the bisection subdivision process. Then we have*

$$\delta(H_{j_q}) \xrightarrow[q\to\infty]{} 0.$$

*Proof.* Let $H_{j_q} = [\underline{a}, \underline{b}] = [a_1, b_1] \times \ldots, \times [a_n, b_n]$.

Then we have

$$\begin{aligned}\delta(H_{j_q}) &= \left(\sum_{i=1}^{n} (b_i - a_i)^2\right)^{\frac{1}{2}} \\ &\leq (n\|b - a\|_\infty)^{\frac{1}{2}} \\ &= \sqrt{n}\|b - a\|_\infty.\end{aligned}$$

$\|b - a\|_\infty$ is halved after at most $n$ subdivision steps, i.e., in the worst case, one subdivision in each coordinate direction has to be performed. Thus

$$\delta(H_{j_{q+kn}}) \leq \frac{\sqrt{n}}{2^k}\|b - a\|_\infty$$

and

$$\delta(H_{j_{q+kn}}) \xrightarrow[q+kn\to\infty]{} 0.$$

□

As discussed in [35, p.24], for any decreasing sequence of partition sets generated by an exhaustive subdivision procedure, the sequence converges to a single point, that is we have

$$\lim_{q\to\infty} H_{j_q} = \bigcap_q H_{j_q} = \bar{x}, \quad \bar{x} \in \mathbb{R}^n.$$

The stopping criterion of Algorithm 1 is based on the distance between upper and lower bound. For convergence or termination it is desirable to use a selection rule that (at least occasionally) improves the lower bound.

**Definition 4.11** ([36, Def. IV.6])**.** A selection operation is said to be bound improving if, at least each time after a finite number of steps, $\mathcal{H}_j$ satisfies the relation

$$\mathcal{H}_j \cap argmin\{p_{lb,H} \mid H \in \mathcal{K}_{j-1}\} \neq \emptyset,$$

i.e., at least one partition element where the actual lower bound is attained is selected for further partition in step $j$ of the algorithm.

As described in [36, p. 130], our selection rule is bound improving due to the finiteness of the number of partition sets. After finitely many steps, any set becomes the one with the longest edge.

To show convergence of our method, we need the following consistency property for our bounding method.

**Definition 4.12** ([36, Def. IV.7])**.** A lower bounding operation is called strongly consistent if at every step any undeleted partition element can be further refined and if any infinite decreasing sequence $\{H_{j_q}\}$ of successively refined partition elements possesses a subsequence $\{H_{j_{q'}}\}$ satisfying

$$\overline{H} \cap P \neq \emptyset, \quad p_{lb,H_{j_{q'}}} \xrightarrow[q'\to\infty]{} p_{min,\overline{H}\cap P},$$

where $\overline{H} = \bigcap_q H_{j_q}$.

To proof the strong consistency property for the proposed methods, we make use of the following well-known property of polynomials.

**Lemma 4.13.** *Let $H$ be a bounded set. Then every polynomial is Lipschitzian on $H$. That means there exists a constant $L$ (Lipschitz constant) such that*

$$|f(x) - f(y)| \leq L\|x - y\| \quad \forall x, y \in H.$$

**Lemma 4.14.** *The Bernstein lower bounding method in the Bernstein branch and bound method and the Handelman bounding method in the Handelman branch and bound method are strongly consistent.*

*Proof.* Note that exhaustiveness of the subdivision method implies that for any descending sequence of partition sets $H_{j_q}$, we have $\bigcap_q H_{j_q} = \bar{x}$ for some $\bar{x} \in \mathbb{R}^n$.

Let $\bar{y}$ be a minimizer of $p$ in $H_{j_q}$, that means $p_{min,H_{j_q}} = p(\bar{y})$. Then we have

$$\begin{aligned} & p(\bar{x}) - p_{Ber,H_{j_q}} \\ = \; & p(\bar{x}) - p(\bar{y}) + p(\bar{y}) - p_{Ber,H_{j_q}} \\ \leq \; & L\|\bar{x} - \bar{y}\| + \Delta(d) \end{aligned}$$

for some $L \geq 0$. The last inequality follows from Lemma 4.13 and Theorem 3.14. Both terms vanish as $\delta(H_{j_q}) \to 0$. This proves the statement for the Bernstein case.

The Handelman case follows in a similar way from Theorem 3.15. □

Finally, to assess the performance of the algorithm, the method to detect and discard infeasible sets plays an important role. As discussed in [36, p. 136], the branch and bound procedure has to ensure that the limit of any nested sequence lies in $P$, that is

$$\overline{H} \cap P \neq \emptyset$$

for $\overline{H} = \bigcap_q H_{j_q}$.

**Definition 4.15** ([36, Def. IV.8])**.** The deletion by infeasibility rule used in the branch and bound method is called certain in the limit, if for every infinite decreasing sequence $H_{j_q}$ of successively refined partition elements with limit $\overline{H}$, we have $\overline{H} \cap P \neq \emptyset$.

The Deletion by Infeasibility rule in the Handelman branch and bound method is certain in the limit, since it decides the feasibility question definitely.

In the Handelman branch and bound method, infeasible sets $H \cap P$ can also be discarded in the pruning step, since $p_{lb,H\cap D} = \infty$ for $H \cap D = \emptyset$, as we show in the following lemma. Since the bounding operation is more expensive than the feasibility test, it is advisable to use the Deletion by Infeasibility rule described above.

**Lemma 4.16.** *Let* $P = \{x \in \mathbb{R}^n \mid g_1(x) \geq 0, \ldots, g_k(x) \geq 0\} = \emptyset$*,* $d \geq m$*.*
*Then* $p^{(d)}_{Han,G} = \infty$*.*

*Proof.* It follows from the Farkas Lemma (Theorem 2.4) that any negative number has a positive combination in $g_1, \ldots, g_k$. □

While the strong consistency property from Definition 4.12 is only concerned with the lower bound, the following definition brings the upper bound into play.

**Definition 4.17** ([36, Definition IV.4])**.** A bounding operation is called consistent, if at every step any unfathomed partition element can be further refined, and if any infinitely decreasing sequence $\{H_{j_q}\}$ of successively refined partition elements satisfies

$$\lim_{q\to\infty} (p_{ub,P}(j_q) - p_{lb,H_{j_q}}) = 0.$$

**Lemma 4.18** ([36, Lemma IV.5]). *Suppose that $p$ is continuous and the subdivision procedure is exhaustive. Furthermore, assume that every infinite decreasing sequence $\{H_{j_q}\}$ of successively refined partition elements satisfies $\emptyset \neq H_{j_q} \cap P$.*

*Then every strongly consistent lower bounding operation does yield a consistent bounding operation.*

With the information collected so far, we can prove Theorem 4.7 and Lemma 4.8. They follow directly from the next theorem.

**Theorem 4.19** ([36, Theorem IV.3]). *In the infinite branch and bound procedure, suppose that the bounding operation is consistent and the selection operation is bound improving. Then the procedure is convergent:*

$$\overline{p_{ub,P}} := \lim_{q\to\infty} p_{ub,P}(q) = p_{min,P} = \lim_{q\to\infty} p_{lb,P}(q) =: \overline{p_{lb,P}}.$$

This immediately proofs Theorem 4.7 and Lemma 4.8.

## 4.3 Variations of the Branch and Bound Scheme

The main points of criticism in the use of Handelman's Positivstellensatz for relaxations of polynomial optimization problems are the slow convergence and the high complexity for high relaxation orders. We avoid these problems by incorporating the Positivstellensatz relaxation into a branch and bound scheme.

In this setting a relaxation of low order has to be solved many times. Therefore it is worthwhile to explore ways of speeding up the bounding step and reducing the number of iterations needed. The first is addressed in the next subsection, afterwards we discuss the impact of the selection rule and the upper bounding method on the speed of convergence.

### 4.3.1 Cleaning Redundant Inequalities

The complexity of the linear and semidefinite programs resulting from Positivstellensatz-relaxations depends on three main factors, the number of variables, the degree-bound and the number of constraints. Only the number of constraints varies throughout the process of solving with our branch and bound method. By adding new inequalities in the branch and bound method, the constraint set is modified. In particular, constraints can become redundant in some subdivision during the process. If additional information on the feasible region is available, cuts can even be chosen in a way to reduce the number of inequalities describing the resulting partitioned sets.

In any case, the constraint set should be cleaned from redundant constraints occasionally in the process of the method. This can be done easily in the linear setting.

It is possible to detect if a linear inequality is redundant by solving a linear program. Let $G = \{g_1(x), \dots g_k(x), g_{k+1}\}$ be the constraint set defining a polytope $P$,

i.e., $P = \{x \in \mathbb{R}^n \mid g_i(x) \geq 0,\ i \in [k+1]\}$. To test if the inequality $g_{k+1}$ is redundant, we can solve the following linear program.

$$\begin{aligned} \text{minimize} \quad & g_{k+1}(x) \\ \text{s.t.} \quad & g_i(x) \geq 0,\ i \in [k] \\ & g_{k+1}(x) \geq -1. \end{aligned} \tag{4.2}$$

The inequality $g_{k+1}(x) \geq 0$ is redundant if and only if the optimal value of (4.2) is greater or equal to 0. This is the method described in Fukuda's "FAQ in Polyhedral Computation", [17].

Cleaning a linear program from redundant inequalities is commonly done in linear programming codes as a presolve step. In this case, heuristics are used to efficiently detect (most) redundant inequalities. A survey on common methods can be found in [39].

As Andersen and Andersen state in [2], there clearly is a trade-off between how much redundancy a presolve procedure detects and the time spent in the presolve procedure. The same is of course true for the work spent on cleaning redundant constraints in the branch and bound scheme.

### 4.3.2 Numerical Tests, Selection Rules and Upper Bounding

We tested the method introduced in this chapter on a range of sample problems. For this, we randomly generated polynomial optimization problems in the following manner.

As feasibility regions, we generated polytopes. The coefficients for the linear constraints were drawn from a uniform distribution on the interval $[-10, 10]$. They are sparse in the sense that only roughly 60% of the coefficients (except the constant term) are non-zero. Additionally all variables are constrained to the range $[-20, 20]$. Only nonempty polytopes were kept.

The coefficients of the objective polynomials were drawn form a uniform distribution on the interval $[-1, 1]$, again with roughly 60% non-zero entries.

In our numerical experiments, we witnessed slow convergence for the basic method described earlier in this chapter. This has two main reasons. The first is the breadth first - selection rule, which always chooses the set with the longest edge of a bounding box. This rule leads to a large number of subsets to be explored before an optimal solution is reached. This is a commonly known problem. Strategies that are commonly used in practical applications include depth first and best bound. We implemented the best bound approach, which consistently subdivides the set with the lowest lower bound currently known. This approach, too, may result in a large list of problems, but it is optimal in the sense that it minimizes the number of subproblems decomposed before the optimum is reached (see [37] for a discussion of search strategies). Notice that the best bound strategy is also bound improving by definition. That means the convergence results from above persist in this situation.

| | | | | selection rule | | | |
|---|---|---|---|---|---|---|---|
| | size | | | breadth first | | best bound | |
| no. | n | k | deg(p) | steps | cputime(s) | steps | cputime(s) |
| 1 | 2 | 3 | 2 | 34 | 37.92 | 26 | 35.71 |
| 2 | 2 | 4 | 4 | 71 | 323.93 | 32 | 123.33 |
| 3 | 2 | 5 | 5 | 65 | 1182.04 | 35 | 620.1 |
| 4 | 2 | 6 | 3 | 224 | 718.93 | 25 | 71.21 |
| 5 | 2 | 7 | 3 | 120 | 331.52 | 25 | 70.89 |
| 6 | 3 | 4 | 4 | 90 | 1924.44 | 27 | 255.7 |
| 7 | 3 | 5 | 3 | 246 | 1730.92 | 39 | 120.01 |
| 8 | 3 | 6 | 4 | 310 | 18585.61 | 47 | 708.11 |

Table 4.2: Number of bisections and cputime needed to solve polynomial problems as described in Example 4.20.

**Example 4.20.** We compared the performance of both strategies on a number of small problems, all generated in the manner described above. As lower bounding method, we solved Handelman relaxations of order $m = deg(p)$. The problems were solved up to a certified error of less than $\epsilon = 0.01$. In Table 4.2, we report the number of branching steps (steps) and the cputime in seconds needed for the complete run.

The second great improvements that can be made is in the upper bounding method. In our numerical experiments, we noticed that the lower bounds converge rapidly, while evaluating the objective on a single point in the upper bounding procedure often leeds to a large number of bisections before the point comes close to the optimal point.

There are several ways to attack this problem. The easiest is to increase the number of points on which the objective is evaluated to accomplish a better coverage of the feasible region. For our implementation, we went a more rigorous way.

**Example 4.21.** Again we performed calculations on a number of problems which where generated as described at the beginning of the section. The problems were solved up to a certified error of less than $\epsilon = 0.01$, using Handelman relaxations of order $m = deg(p)$ as lower bounding method. We used a local minimization method for the upper bounding. We use the "fmincon" method included in MATLAB for our numerical tests. With this setting we are able to solve problems of moderate size in acceptable time, see Table 4.3. We also include the problems from Example 4.20 (no. 1-8). Again, "steps" denotes the number of branching operations, "cputime" the cputime in seconds.

We must emphasize at this point that neither the Handelman bounding nor the implementation of the branch and bound scheme were optimized for speed. We

provide these results rather as a proof of concept than as values that should be benchmarked to other methods.

During the computation the number of subproblems stays quite small (usually less than five problems), this is due to the good quality of lower and upper bounds. They enable us to prune a large number of problems and to keep the search space manageable. The small number of branching operations needed underlines this observation.

We believe the methods described in this chapter bear, if efficiently implemented, great potential for the solution of polynomial problems. From an implementation point of view, branch and bound methods receive attention due to their potential of parallelization. See, e.g., the WIKI of the Berkeley Parallel Computing Laboratory [87] for methods to use the inherent structure of branch and bound schemes.

Apart from implementation details, the use of other Positivstellensätze is the next step that should be explored. In Section 5.5.2, we successfully combine the branch and bound approach with a method to certify containment of spectrahedra. See also Chapter 6 for an outlook on related open problems.

| | size | | | best bound, fmincon | |
|---|---|---|---|---|---|
| no. | n | k | deg(p) | steps | cputime(s) |
| 1 | 2 | 3 | 2 | 0 | 1.72 |
| 2 | 2 | 4 | 4 | 0 | 2.47 |
| 3 | 2 | 5 | 5 | 0 | 4.18 |
| 4 | 2 | 6 | 3 | 7 | 23.20 |
| 5 | 2 | 7 | 3 | 0 | 3.32 |
| 6 | 3 | 4 | 4 | 13 | 106.75 |
| 7 | 3 | 5 | 3 | 1 | 6.73 |
| 8 | 3 | 6 | 4 | 0 | 4.61 |
| 9 | 2 | 3 | 5 | 6 | 23.69 |
| 10 | 2 | 3 | 7 | 1 | 10.75 |
| 11 | 2 | 4 | 3 | 18 | 42.10 |
| 12 | 2 | 4 | 7 | 4 | 67.12 |
| 13 | 2 | 5 | 5 | 25 | 172.10 |
| 14 | 2 | 6 | 3 | 0 | 2.40 |
| 15 | 2 | 6 | 5 | 1 | 13.47 |
| 16 | 2 | 7 | 3 | 0 | 3.24 |
| 17 | 2 | 7 | 5 | 4 | 56.37 |
| 18 | 3 | 4 | 2 | 15 | 41.38 |
| 19 | 3 | 5 | 5 | 1 | 42.04 |
| 20 | 3 | 6 | 3 | 7 | 37.87 |
| 21 | 3 | 6 | 5 | 12 | 2566.58 |
| 22 | 3 | 7 | 2 | 1 | 6.49 |
| 23 | 3 | 7 | 4 | 0 | 6.52 |
| 24 | 3 | 8 | 2 | 0 | 3.55 |
| 25 | 3 | 8 | 4 | 23 | 660.58 |
| 26 | 3 | 9 | 2 | 0 | 3.93 |
| 27 | 3 | 9 | 3 | 0 | 5.41 |
| 28 | 3 | 10 | 2 | 0 | 4.34 |
| 29 | 3 | 10 | 3 | 0 | 6.88 |
| 30 | 4 | 5 | 2 | 0 | 3.16 |
| 31 | 4 | 5 | 3 | 23 | 120.67 |
| 32 | 4 | 5 | 4 | 0 | 5.23 |
| 33 | 4 | 6 | 2 | 20 | 59.38 |
| 34 | 4 | 6 | 3 | 0 | 4.30 |
| 35 | 4 | 6 | 4 | 0 | 8.22 |
| 36 | 4 | 7 | 2 | 0 | 4.19 |
| 37 | 4 | 7 | 3 | 0 | 5.38 |

Table 4.3: Number of bisections and cputime needed to solve polynomial problems as described in Example 4.21.

# 5 Containment Problems for Polytopes and Spectrahedra

In this chapter, we study containment problems for polyhedra and spectrahedra. Since polyhedra are special cases of spectrahedra, we can use the following general setup: Given two linear pencils $A(x) \in \mathcal{S}_k[x]$ and $B(x) \in \mathcal{S}_l[x]$, is $S_A \subseteq S_B$?

For polytopes, the computational geometry and computational complexity of containment problems have been studied in detail. See in particular the classifications by Gritzmann and Klee [23, 24, 25]. In Section 5.1, we extend existing complexity classifications for the polyhedral situation to the situation where polytopes and spectrahedra are involved.

To deal with the the situation that the general containment problem for spectrahedra is co-NP-hard, we propose and study a hierarchy of sufficient semidefinite conditions to certify containment. The proposed technique is derived by encoding the containment problem as a polynomial optimization problem and then applying sum of squares relaxations to it; see Section 5.2.

We demonstrate the effectiveness of the approach by comparing it with the sufficient criterion from [28, 40]. In Section 5.3, we show that the new criterion is satisfied whenever the criterion from [28, 40] is satisfied. Furthermore, numerical results for several containment problems are provided in Section 5.4.

All containment criteria discussed so far are sufficient criteria. In Section 5.5, we explore how to proceed if containment is not certified by the criteria. In the special situation where the inner spectrahedron is a polytope, we can apply the branch and bound scheme derived in the last chapter to find a point $x \in S_A \setminus S_B$, certifying non-containment. We propose to use another geometric branch and bound scheme in the general setting (i.e., when $S_A$ is a spectrahedron), which performs well in our numerical experiments.

*Content published in advance.* Some results of this chapter are published or submitted for publication. Section 5.1 can be found in [40]. The core of Sections 5.2 – 5.4 is based on material from the paper [41] which is currently under review. In [41], the hierarchical criterion is described in terms of a moment relaxation. This is the dual point of view to the sum of squares approach discussed here.

## 5.1 Complexity of Containment Problems for Spectrahedra

In this section, we classify the complexity of several natural containment problems for spectrahedra. Our model of computation is the binary Turing machine: polytopes are presented by certain rational numbers, and the size of the input is defined as the length of the binary encoding of the input data (see, e.g., [23]).

For polytopes the computational complexity of containment problems strongly depends on the type of input representations. Recall from Section 2.2.1 that a polytope can be described as the convex hull of its vertices ($\mathcal{V}$-representation) or as the intersection of halfspaces ($\mathcal{H}$-representation). The following result is well-known (see [16, 25]).

**Proposition 5.1.** *Deciding whether a polytope $P$ is contained in a polytope $Q$ can be done in polynomial time in the following cases:*

1. *Both $P$ and $Q$ are $\mathcal{H}$-polytopes,*
2. *both $P$ and $Q$ are $\mathcal{V}$-polytopes, or*
3. *$P$ is a $\mathcal{V}$-polytope while $Q$ is an $\mathcal{H}$-polytope.*

*However, deciding whether an $\mathcal{H}$-polytope is contained in a $\mathcal{V}$-polytope is co-NP-complete. This hardness persists if $P$ is restricted to be the standard cube and $Q$ is restricted to be the affine image of the cross polytope.*

In the next statements, we extend this classification to containment problems involving polytopes and spectrahedra. See Table 5.1 for a summary. Theorems 5.2 and 5.3 give the positive results.

| | $\mathcal{H}$ | $\mathcal{V}$ | $\mathcal{S}$ |
|---|---|---|---|
| $\mathcal{H}$ | P | co-NP-complete | co-NP-hard |
| $\mathcal{V}$ | P | P | P |
| $\mathcal{S}$ | "SDP" | co-NP-hard | co-NP-hard |

Table 5.1: Computational complexity of containment problems, where the rows refer to the inner set and the columns to the outer set. $\mathcal{S}$ abbreviates spectrahedron.

**Theorem 5.2.** *Deciding whether a $\mathcal{V}$-polytope is contained in a spectrahedron can be done in polynomial time.*

*Proof.* Given a $\mathcal{V}$-representation $P = \operatorname{conv}\{v^{(1)}, \dots, v^{(m)}\}$ and a linear matrix pencil $A(x)$, we have $P \subseteq S_A$ if and only if all the points $v^{(i)}$ are contained in $S_A$. Thus, the containment problem is reduced to $m$ tests whether a certain rational matrix is positive semidefinite. This can be decided in polynomial time, as one can compute, for a rational, symmetric matrix $A$, a decomposition $A = UDU^T$ with a diagonal matrix $D$ in polynomial time (see, e.g., [21]). □

Containment questions for spectrahedra are connected to feasibility questions of semidefinite programs in a natural way. As discussed in Section 2.2.2, the complexity of semidefinite feasibility problems is not known. Consequently, the following statement on containment of a spectrahedron in an $\mathcal{H}$-polytope does not give a complete answer concerning polynomial solvability of these containment questions in the Turing machine model. If the additional inequalities were non-strict, then we had to decide a finite set of problems from the complement of the class SDFP.

**Theorem 5.3.** *The problem of deciding whether a spectrahedron is contained in an $\mathcal{H}$-polytope can be formulated by the complement of semidefinite feasibility problems (involving also strict inequalities), whose sizes are polynomial in the description size of the input data.*

*Proof.* Let $A(x)$ be a linear matrix pencil and $P = \{x \in \mathbb{R}^n : b + Bx \geq 0\}$ with $B \in \mathbb{Q}^{m\times n}$ be an $\mathcal{H}$-polytope. For each $i \in \{1, \ldots, m\}$ incorporate the linear condition $b_i + \sum_{j=1}^{n} b_{ij}x_j < 0$ into the linear pencil $A(x)$. If one of the resulting $m$ ("semi-open") spectrahedra is nonempty then $S_A \not\subseteq P$. □

The positive results in Theorems 5.2 and 5.3 are contrasted by the following hardness results.

**Theorem 5.4.**

1. *Deciding whether a spectrahedron is contained in a $\mathcal{V}$-polytope is co-NP-hard.*
2. *Deciding whether an $\mathcal{H}$-polytope or a spectrahedron is contained in a spectrahedron is co-NP-hard. This hardness statement persists if the $\mathcal{H}$-polytope is a standard cube or if the outer spectrahedron is a ball.*

*Proof.* Deciding whether a spectrahedron $S_A$ is contained in a $\mathcal{V}$-polytope is co-NP-hard since already deciding whether an $\mathcal{H}$-polytope is contained in a $\mathcal{V}$-polytope is co-NP-hard by Proposition 5.1.

Concerning the second statement, co-NP-hardness of containment of $\mathcal{H}$-polytopes in spectrahedra follows from Ben-Tal and Nemirovski [6, Proposition 4.1], who use a reduction from the maximization of a positive semidefinite quadratic form over the unit cube.

For the co-NP-hardness of containment of an $\mathcal{H}$-polytope in a ball, we provide a reduction from the NP-complete 3-satisfiability problem (3-SAT [10]): Does a given Boolean formula $\Phi$ over the variables $z_1, \ldots, z_n$ in conjunctive normal form, where each clause has at most 3 literals, admit an assignment that evaluates TRUE?

The $2^n$ possible assignments $\{\text{FALSE}, \text{TRUE}\}^n$ for $z_1, \ldots, z_n$ can be identified with the vertices of an $n$-dimensional cube $[-1, 1]^n$. Let $B$ be a ball (which is a spectrahedron), such that the vertices of $[-1, 1]^n$ just "peak" through its boundary sphere $S$. Precisely (assuming w.l.o.g. $n \geq 2$), choose the radius $r$ of $B$ such that

$$\left(\frac{1}{6}\right)^2 + \left(\sqrt{n} - \frac{1}{6}\right)^2 \;<\; r^2 \;<\; n\,.$$

Note that such a radius can be determined in polynomial time and size.

For the definition of the $\mathcal{H}$-polytope $P$, we start from the $\mathcal{H}$-representation $\{x \in \mathbb{R}^n : -1 \leq x_i \leq 1, 1 \leq i \leq n\}$ of $[-1,1]^n$ and add one inequality for each clause of $\Phi$. Let $\mathcal{C} = \mathcal{C}_1 \vee \cdots \vee \mathcal{C}_m$ be a 3-SAT formula with clauses $\mathcal{C}_1, \ldots, \mathcal{C}_m$. Denote by $\overline{z_i}$ the complement of a variable $z_i$, and define the literals $z_i^1 := z_i$, $z_i^0 := \overline{z_i}$. If the clause $\mathcal{C}_i$ is of the form $\mathcal{C}_i = z_{i_1}^{e_{i_1}} \vee z_{i_2}^{e_{i_2}} \vee z_{i_3}^{e_{i_3}}$ with $e_{i_1}, e_{i_2}, e_{i_3} \in \{0,1\}$, then add the inequality

$$(-1)^{e_{i_1}} x_{i_1} + (-1)^{e_{i_2}} x_{i_2} + (-1)^{e_{i_3}} x_{i_3} \;\leq\; 1\,.$$

If $P \subseteq B$, then, by the choice of $r$, none of the points in $\{-1,1\}^n$ can be contained in $P$ and thus there does not exist a valid assignment for $\Phi$. Conversely, assume that $P$ is not contained in $B$. Let $p \in P \setminus B \subseteq [-1,1]^n$. We claim that componentwise rounding of $p$ yields an integer point $p' \in \{-1,1\}^n$ satisfying all defining inequalities of $P$. To see this, first note that by the choice of the radius of $B$, the components $p_i$ of $p$ differ at most $\varepsilon < \frac{1}{3\sqrt{2}} < \frac{1}{3}$ from either $-1$ or 1.

In order to inspect what happens to the inequalities when rounding, assume without loss of generality that the inequality is of the form $x_1 + x_2 + x_3 \geq -1$. We assume a rounded vector $p'$ does not satisfy the inequality, even though $p$ does:

$$p'_1 + p'_2 + p'_3 \;<\; -1\,, \quad \text{but } p_1 + p_2 + p_3 \;\geq\; -1\,. \tag{5.1}$$

Since $p' \in \{-1,1\}^n$, (5.1) implies $p'_1 = p'_2 = p'_3 = -1$. Hence, at least one of $p_1, p_2$ and $p_3$ differs from either $-1$ or 1 by more than $1/3$, which is a contradiction. This completes the reduction from 3-SAT.

Finally, deciding whether a spectrahedron $S_A$ is contained in a spectrahedron $S_B$ is co-NP-hard, since already deciding whether an $\mathcal{H}$-polytope is contained in a spectrahedron is co-NP-hard. □

## 5.2 A (Sufficient) Semidefinite Hierarchy

Let $A(x) \in \mathcal{S}_k[x]$ and $B(x) \in \mathcal{S}_l[x]$ be two linear pencils. In this section, we provide an optimization formulation to decide the question of whether the spectrahedron $S_A$ is contained in $S_B$. Using a polynomial matrix inequality formulation of the containment problem, we first deduce a sufficient semidefinite hierarchy and prove the convergence of the hierarchy (Theorem 5.7).

### 5.2.1 An Optimization Approach to Decide Containment of Spectrahedra

Clearly, $S_A$ is contained in $S_B$ if and only if $A(x) \succeq 0$ implies the positive semidefiniteness of $B(x)$. By definition, $B(x) \succeq 0$ for arbitrary but fixed $x \in \mathbb{R}^n$ is equivalent to the non-negativity of the polynomial $z^T B(x) z$ in the variables $z = (z_1, \ldots, z_l)$. Thus, $S_A$ is contained in $S_B$ if and only if the infimum $\mu$ of

the degree 3 polynomial $z^T B(x)z$ in $(x, z)$ over the spectrahedron $S_A \times \mathbb{R}^l$ is non-negative. Imposing a normalization condition on $z$, we arrive at the following formulation.

**Proposition 5.5.** *Let $A(x) \in \mathcal{S}_k[x]$ and $B(x) \in \mathcal{S}_l[x]$ be linear pencils with $S_A \neq \emptyset$, and let $g_r(z) = z^T z - r^2$, $g^R(z) = R^2 - z^T z$ for arbitrary but fixed $0 < r \leq R$. For the polynomial optimization problem*

$$\begin{aligned} \text{minimize} \quad & z^T B(x) z \\ \text{s.t.} \quad & G_A(x,z) := \operatorname{diag}(A(x), g_r(z), g^R(z)) \succeq 0 \end{aligned} \tag{5.2}$$

*with optimal value $\mu$, the following implications are true,*

$$\begin{aligned} \mu > 0 \ &\Rightarrow S_A \subseteq \operatorname{int} S_B, \\ \mu = 0 \ &\Rightarrow S_A \subseteq S_B, \\ \mu < 0 \ &\Leftrightarrow \exists x \in S_A : B(x) \not\succeq 0. \end{aligned}$$

*If the pencil $B(x)$ is reduced in the sense of Proposition 2.6, $\mu = 0$ implies that the spectrahedra touch at the boundary.*

A natural choice of the parameters $r$ and $R$ is to set both to 1. In this case, the optimal value of the optimization problem equals the smallest eigenvalue of any matrix in the set $\{B(x) \mid x \in S_A\}$. Other choices result in an optimal value that is scaled by $R^2$ in the case $\mu < 0$ and by $r^2$ in the case $\mu > 0$. As our numerical computations in Section 5.4 show, the problem, or, more precisely, its relaxation defined in Section 5.2.2 is numerically ill-conditioned if we chose $r = R$ and becomes more tractable for $r < R$.

In applications, it is advisable to use reduced pencils. The reduced pencil can be computed by the methods in [20] and makes the numerical computations described below better conditioned. Not only do we expect a strictly positive objective value whenever $S_A \subset S_B$, the reduced pencil is also of smaller size.

*Proof (of Proposition 5.5).* Denote by $\mathbb{T} = \mathbb{T}_{r,R}(0) = \{z \in \mathbb{R}^l \mid r^2 \leq z^T z \leq R^2\}$ the annulus defined by the constraints $g_r(z) \geq 0,\ g^R(z) \geq 0$.

We first observe that the existence of an $x \in S_A$ and $z \in \mathbb{R}^l$ with $z^T B(x) z < 0$ implies the existence of a point $z' := R \cdot \frac{z}{\|z\|} \in \mathbb{T}$ with $\|z'\| = R$ and $z'^T B(x) z' < 0$, and thus $(x, z')$ lies in the product of the spectrahedron $S_A$ and the annulus $\mathbb{T}$.

If $\mu \geq 0$, then clearly $S_A \subseteq S_B$. To deduce the case $\mu > 0$, observe that the boundary $\partial S_B$ of $S_B$ is contained in the set

$$\{x \in \mathbb{R}^n \mid B(x) \succeq 0,\ z^T B(x) z = 0 \text{ for some } z \in \mathbb{T}\}.$$

Hence, if the boundaries of $S_A$ and $S_B$ contain a common point $\bar{x}$, then there exists some $\bar{z}$ such that the objective value of $(\bar{x}, \bar{z})$ is zero. □

### 5.2.2 Derivation of the Hierarchy Using PMI Methods

Using the framework of sum of squares relaxations for polynomial matrix inequalities introduced in Section 2.3.3, we consider the following semidefinite hierarchy as a relaxation to problem (5.2).

$$\begin{aligned} \text{maximize} \quad & t \\ \text{s.t.} \quad & z^T B(x) z - \langle S(x,z), G_A(x,z) \rangle - t \ \in \Sigma_{n+l} \\ & S(x,z) \in \Sigma_{n+l,d}^{(k+2)\times(k+2)}. \end{aligned} \tag{5.3}$$

Denote by $\mu_{\text{sos}}(d)$ the optimal value of the $d$-th relaxation step. Since a sum of squares polynomial necessarily has even degree and the polynomial $z^T B(x) z$ has degree 3, the smallest sensible relaxation order is $d = 2$, which we call the initial relaxation order for the problem.

By increasing $d$, additional constraints are added, which implies the following corollary.

**Corollary 5.6.** *The sequence $\mu_{\text{sos}}(d)$ for $d \geq 2$ is monotone increasing. If for some $d^*$ the condition $\mu_{\text{sos}}(d^*) \geq 0$ is satisfied, then $S_A \subseteq S_B$.*

That is, for any $d$, the condition $\mu_{\text{sos}}(d) \geq 0$ provides a sufficient criterion for the containment $S_A \subseteq S_B$. In the case when the inner spectrahedron $S_A$ is bounded, the sequence of relaxations is not only monotone increasing, but also converges to the optimal value of the original polynomial optimization problem (5.2), as the next theorem shows.

**Theorem 5.7.** *Let $A(x) \in \mathcal{S}_k[x]$ be a linear pencil such that the spectrahedron $S_A$ is bounded. Then the optimal value of the sum of squares relaxation* (5.3) *converges from below to the optimal value of the polynomial optimization problem* (5.2)*, i.e., $\mu_{\text{sos}}(d) \uparrow \mu$ as $d \to \infty$.*

*Proof.* By Proposition 2.21, it suffices to show that there exists an sos-polynomial $s(x,z) \in \mathbb{R}[x,z]$ and an sos-matrix $S(x,z) \in \mathcal{S}_{k+2}[x,z]$ defining a polynomial $p(x,z) = s(x,z) + \langle S(x,z), G_A(x,z) \rangle$ such that the set $\{(x,z) \in \mathbb{R}^{n+l} \mid p(x,z) \geq 0\}$ is compact. Define the quadratic module

$$M_A = \left\{ t(x) + \langle A(x), T(x) \rangle \ \mid t(x) \in \Sigma,\ T(x) \in \Sigma^{k\times k} \right\}.$$

As shown in [42, Lemma 4.4.1], the boundedness of $S_A$ is equivalent to the fact that the quadratic module $M_A$ is Archimedean, i.e., there exists a positive integer $N \in \mathbb{N}$ such that $N - x^T x \in M_A$. Thus, by the definition of the quadratic module $M_A$, there exists an sos-polynomial $t(x) \in \mathbb{R}[x]$ and an sos-matrix $T(x) \in \mathcal{S}_k[x]$ such that

$$N - x^T x = t(x) + \langle T(x), A(x) \rangle .$$

Define $s(x,z) = t(x)$ and $S(x,z) = \operatorname{diag}(T(x), 0, 1)$. Both have the sos-property. Indeed, if $T(x) = U(x)U(x)^T$ is an sos-decomposition of $T(x)$, then $S(x,z) = \operatorname{diag}(T(x),0,1) = \operatorname{diag}(U(x),0,1)\operatorname{diag}(U(x)^T,0,1)$ is one of $S(x,z)$. We get

$$p(x,z) = N - x^T x + R^2 - z^T z = s(x,z) + \langle S(x,z), G_A(x,z)\rangle\,.$$

Since this polynomial defines the ball of radius $N + R^2$ centered at the origin, $\mathbb{B}_{N+R^2}(0) \subset \mathbb{R}^{n+l}$, the level set is compact. □

*Remark* 5.8. Computing a certificate $N$ from the proof of the theorem can again be done by the polynomial program (5.2) and its relaxation (5.3).

## 5.3 Exact Cases

In this section, we first review the containment criterion based on a semidefinite feasibility problem, that was studied in [28, 40]. We then prove that the sufficient criteria coming from our hierarchy of relaxations are at least as strong as this feasibility criterion by showing that feasibility of the criterion implies $\mu_{sos}(2) \geq 0$. From this relation, we get that in some cases already the initial relaxation step gives an exact answer to the containment problem; see Corollaries 5.12 and 5.13.

The semidefinite feasibility criterion described in [28, 40] is based on another sufficient condition for the pencil $B(x)$ to be positive semidefinite whenever $A(x)$ is positive semidefinite.

**Proposition 5.9.** *[40, Theorem 4.3] If the semidefinite feasibility problem*

$$C = (C_{ij})_{i,j=1}^{k} \succeq 0 \text{ and } B_p = \sum_{i,j=1}^{k} a_{ij}^p C_{ij} \quad \text{for } p = 0,\ldots,n \tag{5.4}$$

*has a solution, then* $S_A \subseteq S_B$.

In terms of the linear pencils, the previous proposition states that the pencil $B(x) = \sum_{ij=1}^{k} (A(x))_{ij} C_{ij}$ is positive semidefinite if $A(x)$ and $C$ are positive semidefinite. The proof in [40] is by elementary calculation.

The matrix $C$ can also be interpreted as a linear map from the pencil $A(x)$ to the pencil $B(x)$. In this setting, the feasibility problem above is feasible if and only if the map is completely positive. Further details on map positivity and the connection to the containment relaxations are given in [41].

As we will see next, positive semidefiniteness of the matrix $C$ is not only a sufficient condition for containment and thus for the non-negativity of the polynomial optimization problem in Proposition 5.5, but also for its relaxation (5.3).

We show the following result:

**Theorem 5.10.** *If the SDFP* (5.4) *has a solution, then the optimal value* $\mu_{\mathrm{sos}}(2)$ *of the initial relaxation in* (5.3) *is non-negative.*

*Proof.* Assume $C \succeq 0$ is a solution to the SDFP. We claim that the following matrix is a solution to (5.3) with optimal value $\mu_{\text{sos}}(d) = 0$ for $d \geq 2$

$$S(x,z) = \begin{pmatrix} \left(z^T C_{ij} z\right)_{i,j=1}^{k} & & \\ & 0 & \\ & & 0 \end{pmatrix}.$$

Let us first show that $z^T B(x) z - \langle S(x,z), G_A(x,z)\rangle - 0 \in \Sigma_{k+l,d}$ by showing that the first two terms are equal. This can be seen using the equality from system (5.4).

$$\begin{aligned} z^T B(x) z &= z^T \left( \sum_{i,j=1}^{k} C_{ij} A(x)_{ij} \right) z = \sum_{i,j=1}^{k} z^T C_{ij} z \, A(x)_{ij} \\ &= \left\langle \left(z^T C_{ij} z\right)_{i,j=1}^{k}, A(x) \right\rangle = \langle S(x,z), G_A(x,z)\rangle . \end{aligned}$$

It remains to show that $S(x,z) \in \Sigma_{n,d}^{k+2\times k+2}$. We can ignore the zero-entries in the lower right corner and focus on the submatrix $\left(z^T C_{ij} z\right)_{i,j=1}^{k}$. Denote by $\mathbb{0}_l$ the all-zero vector in $\mathbb{R}^l$ and by $Z$ the matrix

$$Z := \begin{pmatrix} z & \mathbb{0}_l & \cdots & \mathbb{0}_l \\ \mathbb{0}_l & z & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbb{0}_l \\ \mathbb{0}_l & \cdots & \mathbb{0}_l & z \end{pmatrix}.$$

With this notation, we have

$$\left(z^T C_{ij} z\right)_{i,j=1}^{k} = Z^T C Z = Z^T L^T L Z,$$

where $C = L^T L$ is the Cholesky decomposition of $C$, which exists since $C$ is positive semidefinite by assumption. This shows that $S(x,z)$ has a decomposition of the form $S(x,z) = T(x,z)^T T(x,z)$ with $T(x,z) = LZ$. Therefore it is a sum of squares matrix. □

*Remark* 5.11. (1) The reverse implication in Theorem 5.10 is not always true. Example 5.15 serves as a counterexample.
(2) By inspecting the proof, we see that it can be reasonable to use a lower relaxation order than what we described as initial relaxation order above. We can not decrease the degree of the representation in the $z$-variables, but the matrix $S(x,z)$ can be constrained to be a sum of squares matrix only in the $z$-variables, independent of $x$. This may be an option for very large systems where the initial relaxation is too expensive to solve.

Using Theorem 5.10, we can extend the exactness results from [40] to the hierarchy (5.3), i.e., in some cases already the initial relaxation is not only a sufficient condition but also necessary for containment. These results rely on the specific pencil representation of the given spectrahedra. Before stating the results, we have to agree on a consistent representation. Recall the normal forms of the polyhedron, the ball and the ellipsoid as a spectrahedron given in Section 2.2.2.

Then we can state the following exactness results.

**Corollary 5.12.** *Let $A(x) \in \mathcal{S}_k[x]$ and $B(x) \in \mathcal{S}_l[x]$ be monic linear pencils. In the following cases, the initial relaxation step ($d = 2$) in (5.3) certifies containment of $S_A$ in $S_B$.*

1. *if $A(x)$ and $B(x)$ are normal forms of ellipsoids (both centrally symmetric, axis-aligned semiaxes),*
2. *if $A(x)$ and $B(x)$ are normal forms of a ball and an $\mathcal{H}$-polyhedron, respectively,*
3. *if $B(x)$ is the normal form of a polytope,*
4. *if $\widehat{A}(x)$ (see (2.10)) is the extended form of a spectrahedron and $B(x)$ is the normal form of a polyhedron.*

*Proof.* Follows directly from [40, Theorem 4.8] and Theorem 5.10. □

The second exactness result states that the initial relaxation step can always certify containment of a scaled situation.

**Corollary 5.13.** *Let $A(x) \in \mathcal{S}_k[x]$ and $B(x) \in \mathcal{S}_l[x]$ be monic linear pencils such that $S_A$ is bounded. Then there exists $\nu > 0$ such that the initial relaxation step certifies $\nu S_A \subseteq S_B$, where $\nu S_A = \{x \in \mathbb{R}^n : A^\nu(x) := A(\frac{x}{\nu}) \succeq 0\}$ is the scaled spectrahedron.*

*Proof.* This follows from [40, Proposition 6.2] and Theorem 5.10. □

## 5.4 Numerical Experiments

While the complexity of the containment question for spectrahedra is co-NP-hard in general, the relaxation techniques introduced above give a practical way of certifying containment. We implemented the hierarchy and applied it to several examples. The criterion performs well already for relaxation orders as low as $d = 2, 4$, as we will witness throughout this section.

We start by reviewing an example from [40] in Section 5.4.1, showing that the new hierarchical relaxation indeed outperforms the feasibility criterion 5.4. We then give an overview on the performance of the relaxation on some more examples.

In Section 5.4.2, to assess the performance of the algorithms, we compare results as well as running times of the algorithms on randomly generated pencils of varying sizes.

Throughout this section, we use the following notation. As before, integer $n$ stands for the number of variables in the pencils, $k$ and $l$ for the size of the pencil

$A(x)$ and $B(x)$, respectively. For monic pencils, we examine $\nu$-scaled spectrahedra $\nu S_A$ as defined in Corollary 5.13. We denote the (numerical) optimal value of the sum of squares relaxation (5.3) of order $d$ by $\mu_{\text{sos}}(d)$. In the tables, "sec" states the time in seconds for setting up the problem in YALMIP and solving it in MOSEK.

Unless stated otherwise, the inner radius is set to $r = 1$ and the outer radius to $R = 2$ in the relaxation (5.3).

### 5.4.1 Numerical Computations

We review the example of containment of two disks from [40]. The complete positivity criterion from that work certifies the containment only if the disk on the inside is scaled small enough. Theorem 5.10 shows that any containment certified by the feasibility criterion is certified by the hierarchical relaxation. In the following example we go one step further, showing that the latter performs strictly better than the feasibility criterion already in small relaxation orders.

**Example 5.14.** Consider the monic linear pencils $A^\nu(x) = I_3 + x_1\frac{1}{\nu}(E_{1,3}+E_{3,1}) + x_2\frac{1}{\nu}(E_{2,3}+E_{3,2}) \in \mathcal{S}_3[x]$ with parameter $\nu > 0$ and $B(x) = I_2 + x_1(E_{1,1} - E_{2,2}) + x_2(E_{1,2} + E_{2,1}) \in \mathcal{S}_2[x]$. The spectrahedra defined by the pencils are the disk of radius $\nu > 0$ centered at the origin, $\nu S_A = \mathbb{B}_\nu(0)$, and the unit disk $S_B = \mathbb{B}_1(0)$, respectively. Clearly, $\nu S_A \subseteq S_B$ if and only if $0 < \nu \leq 1$. In particular, for $\nu = 1$, both pencils define the unit disk $\mathbb{B}_1(0) = S_A = S_B$.

In [40, Section 6.1], it is shown that the feasibility criterion for the containment problem $\nu S_A \subseteq S_B$ is satisfied if $0 < \nu \leq \frac{1}{2}\sqrt{2}$. Remarkably, the performance of relaxation (5.3) depends on the choice of the parameters $r$ and $R$. Table 5.2 contrasts the results of the sums of squares relaxation with parameters $r = 1, R = 2$ with the results of the feasibility criterion for the problem $\nu S_A \subseteq S_B$. Our numerical computations show that the semidefinite relaxation of order $d = 2$ certifies the same cases as the feasibility criterion. For $d = 4$, we have exactness of the criterion.

When choosing $r = R = 1$, the semidefinite relaxation (5.3) is exact already for relaxation order $d = 2$ and returns the same optimal values as for relaxation order $d = 4$. This choice of parameters however leads to numerical problems in the solver occasionally. Furthermore, the example of the two disks is the only one we have found, where results for orders $d = 2$ and $d = 4$ differ if $r$ and $R$ are chosen distinct. In all other examples, results seem to be exact already for $d = 2$. Therefore, we advise to use $r = 1$ and $R = 2$ in general applications.

In the next example, we examine the containment of a ball in an elliptope. The elliptope is a nice example of a spectrahedron that is described by a pencil consisting of very sparse matrices. While the pencil is of small size, it is occupied by a large number of variables.

**Example 5.15.** For this example, the pencil description of the ball is as in (2.12). The *elliptope* can be described as the positivity domain of a symmetric pencil

| $\nu$ | SDFP (5.4) | $\mu_{\text{sos}}(2)$ | sec | $\mu_{\text{sos}}(4)$ | sec |
|---|---|---|---|---|---|
| 0.7 | feasible | 0.010 | 0.27 | 0.3 | 0.41 |
| 0.707 | feasible | 0.000 | 0.27 | 0.293 | 0.41 |
| $1/\sqrt{2}$ | feasible | 0.000 | 0.27 | 0.293 | 0.40 |
| 0.708 | infeasible | -0.001 | 0.27 | 0.292 | 0.41 |
| 0.8 | infeasible | -0.066 | 0.27 | 0.2 | 0.41 |
| 1 | infeasible | -0.207 | 0.27 | 0 | 0.42 |
| 1.1 | infeasible | -0.278 | 0.27 | -0.1 | 0.42 |

Table 5.2: Disk $\nu S_A$ in disk $S_B$ for two different representations and various radii $\nu$ of the inner disk as described in Example 5.14. Parameters are chosen as $r = 1, R = 2$.

| $n$ | $k$ | $l$ | SDFP (5.4) | $\mu_{\text{sos}}(2)$ | sec |
|---|---|---|---|---|---|
| 3 | 4 | 3 | feasible | 0.293 | 0.30 |
| 6 | 7 | 4 | feasible | 0.134 | 0.86 |
| 10 | 11 | 15 | feasible | 0.106 | 5.65 |
| 15 | 16 | 6 | infeasible | 0.087 | 46.86 |

Table 5.3: Computational test of containment of ball in elliptope as described in Example 5.15.

with ones on the diagonal and distinct variables in the remaining positions; see [8, Section 2.1.3].

As exhibited in Table 5.3, the ball of radius $\frac{1}{2}$ in Dimensions $n = 3, 6, 10$ and 15 is contained in the elliptope of the respective dimension. The computational time grows in the number of variables, but even dimensions as high as 15 are in the scope of desktop computers if the size $l$ of the pencil $B(x)$ is moderate.

Note that the SDFP (5.4) is feasible for $(n, k, l) = (10, 11, 5)$ but not feasible for $(n, k, l) = (15, 16, 6)$. Thus, for $(n, k, l) = (15, 16, 6)$, this example serves as a counterexample for the reverse statement of Theorem 5.10.

### 5.4.2 Randomly Generated Spectrahedra

We applied both the hierarchical criterion (5.3) and the feasibility criterion (5.4) to several instances of linear pencils with random entries.

For the experiments in this section, we generate coefficient matrices $A_1, \ldots, A_n$ by assigning random numbers to the off-diagonal entries of the matrices. Numbers are drawn from a uniform distribution on $[-1, 1]$. The generated matrices are sparse in the sense that roughly 35% of the off-diagonal entries are nonzero. The matrix for the constant term $A_0$ is generated in the same way, but features ones on the diagonal. This choice leads to bounded spectrahedra in most cases, namely

| | size | | | feasibility | objective value | | sec | |
|---|---|---|---|---|---|---|---|---|
| no. | $n$ | $k$ | $l$ | SDFP(5.4) | $\mu_{\text{sos}}(2)$ | $\mu_{\text{sos}}(4)$ | $\mu_{\text{sos}}(2)$ | $\mu_{\text{sos}}(4)$ |
| 1 | 2 | 4 | 4 | infeasible | 0.330 | 0.330 | 0.91 | 3.06 |
| 2 | 2 | 6 | 4 | feasible | 1.459 | 1.459 | 0.35 | 3.66 |
| 3 | 2 | 4 | 6 | infeasible | -2.009 | -2.009 | 0.57 | 31.60 |
| 4 | 2 | 6 | 6 | infeasible | -0.209 | -0.209 | 0.65 | 45.84 |
| 5 | 3 | 4 | 4 | infeasible | 0.156 | 0.156 | 0.39 | 7.39 |
| 6 | 3 | 6 | 4 | infeasible | 0.332 | 0.332 | 0.43 | 9.86 |
| 7 | 3 | 4 | 6 | infeasible | -6.918 | -6.918 | 0.78 | 103.91 |
| 8 | 3 | 6 | 6 | infeasible | 0.028 | 0.028 | 0.87 | 97.28 |
| 9 | 4 | 4 | 4 | infeasible | -3.164 | -3.164 | 0.50 | 29.46 |
| 10 | 4 | 6 | 4 | infeasible | 0.593 | 0.593 | 0.58 | 34.83 |
| 11 | 4 | 4 | 6 | infeasible | -0.938 | -0.938 | 1.14 | 285.38 |
| 12 | 4 | 6 | 6 | infeasible | -0.251 | -0.251 | 1.23 | 368.22 |

Table 5.4: Computational test of containment of randomly generated spectrahedra as described in Example 5.16.

when the matrices $A_0, \ldots, A_k$ are linearly independent. Unbounded spectrahedra and spectrahedra without interior are discarded.

The pencil of the second spectrahedron $S_B$ is generated in the same way, except that the diagonal entries of $B_0$ are chosen larger. This has the effect that the corresponding spectrahedra are scaled and the containment $S_A \subseteq S_B$ is more likely to happen.

**Example 5.16.** We apply the hierarchies to a range of problems with varying dimensions and pencil sizes as reported in Table 5.4. To illustrate the approach, we provide the pencils for experiment no. 1 below.

$$A(x) = \begin{bmatrix} 1 & 0.2528x_1 + 0.3441x_2 & 0 & 0 \\ 0.2528x_1 + 0.3441x_2 & 1 & 0 & -0.1314x_1 \\ 0 & 0 & 1 & 0.7969x_2 \\ 0 & -0.1314x_1 & 0.7969x_2 & 1 \end{bmatrix},$$

$$B(x) = \begin{bmatrix} 2 & 0.8454 & 0 & 0 \\ 0.8454 & 2 & -0.2489x_1 - 0.4063x_2 & 0 \\ 0 & -0.2489x_1 - 0.4063x_2 & 2 & 0.3562x_1 \\ 0 & 0 & 0.3562x_1 & 2 \end{bmatrix}.$$

For this experiment with randomly generated matrices, the truth value of the containment question is unknown a priori. In the case of non-negative objective values, our criterion yields a certificate for the containment. For negative objective values, we inspected plots of the spectrahedra to check appropriateness of the

criterion. Plots of the spectrahedra from the two-dimensional experiments no. 1–4 are shown in Figure 5.1.

In cases of higher dimension ($n > 3$), we examined projections of the spectrahedra. See Figure 5.2 for projections of the spectrahedra from experiment no. 12 to different planes. The small negative objective value reported in Table 5.4 suggests that there is only a small overlap of $S_A$ over the boundary of $S_B$. Indeed, the projections to the coordinate planes suggest that $S_A$ is contained in $S_B$. But when projecting to the plane spanned by $0.3x_1 + x_2$ and $x_3$, we see that the spectrahedra are not contained.

See also Section 5.5 for an algorithmic method to certify non-containment.

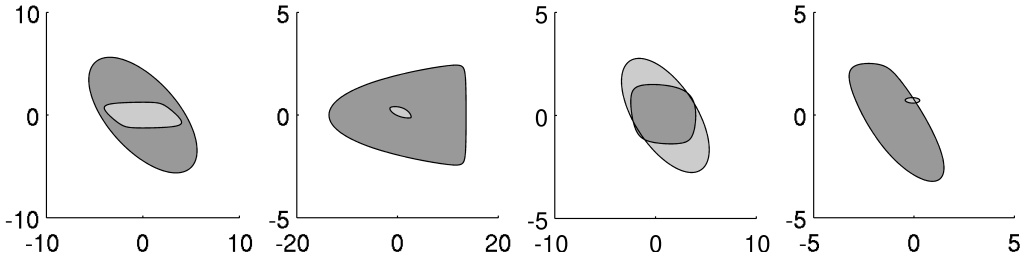


Figure 5.1: Spectrahedra of experiments no. 1–4 from Table 5.4. $S_A$: light grey, $S_B$: dark grey.

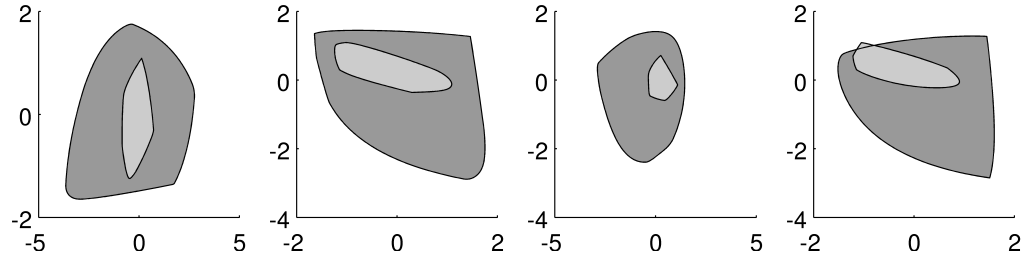


Figure 5.2: Projections of the 4-dimensional spectrahedron no. 12 from Table 5.4. $S_A$: light grey, $S_B$: dark grey. Projections to planes spanned by $(x_1, x_2)$, $(x_2, x_3)$, $(x_3, x_4)$ and $(0.3x_1 + x_2, x_3)$

In all cases we examined, the results from the hierarchical criterion correspond with the expectations we had from inspecting the plots. Remarkably, for the randomly generated spectrahedra, the results of the relaxations match closely across different relaxation orders and across the two approaches discussed. This suggests that the criteria perform well in generic cases.

As expected, running times increase quickly with growing dimension $n$ and with an increase in the sizes $k$ and $l$ of the pencils $A(x)$ and $B(x)$.

This can be explained by inspecting the size of the resulting semidefinite program. As explained in Theorem 2.22, the two sum of squares constraints can be

modeled by semidefiniteness constraints in $\mathcal{S}_{k\binom{n+l+d-1}{d}} \times \mathcal{S}_{\binom{n+l+d+1}{d+2}}$. To model the first constraint in (5.3), we need $\binom{n+l+2d-1}{2d}$ equality constraints in addition. The number of variables is given by

$$\begin{aligned} 1 + \frac{1}{2}k\binom{n+l+d-1}{d}\left[k\binom{n+l+d-1}{d}+1\right] \\ + \frac{1}{2}\binom{n+l+d}{d+1}\left[\binom{n+l+d}{d+1}+1\right]. \end{aligned} \tag{5.5}$$

See also [80, Section 5] for a similar discussion.

While $k$ enters the formula only quadratically, $l$ appears in the binomial coefficients. As one would expect, increasing $l$ has much more effect on the running time than increasing $k$.

Standard semidefinite programming duality theory tells us that the dual problem has $\binom{n+l+2d-1}{2d}$ variables and the number of equality constraints is given by equation (5.5). We interpreted the primal problem as a truncated sum of squares problem. In a similar way, we can interpret the dual problem as a truncated moment relaxation, see [31, 41].

We implemented this dual approach and again performed numerical experiments.

**Example 5.17.** We apply the hierarchies to a range of problems with varying dimensions and pencil sizes as reported in Table 5.5. By $\mu_{\text{mom}}(d)$, we denote the optimal value of the dual problem to the $d$-th sum of squares relaxation. We fix the spectrahedron $S_B$ for each dimension $n$ and vary the size of the (randomly generated) pencil describing $S_A$ to see the impact of changing the size $k$.

The running time for the moment approach is oftentimes shorter than for the sum of squares approach. It is a common phenomenon that a large number of variables has a greater impact on the complexity of a semidefinite program than the number of constraints. It should be kept in mind that neither the sum of squares implementation nor the moment implementation have been optimized for speed and the running times in the table can only give rough estimates.

## 5.5 Certifying Non-Containment

The method introduced in Section 5.2 yields a lower bound on the optimal value of the polynomial optimization problem (5.2). The procedure can only verify containment of a spectrahedron $S_A$ in another one $S_B$, if this lower bound is non-negative. If the lower bound is negative, the containment question remains undecided.

In this section, we discuss methods to find a point $x \in S_A \setminus S_B$ that serves as a certificate for non-containment. These methods are based on the branch and bound scheme discussed in the last chapter.

Let us first look at the case where $S_A$ is a polytope before discussing the general case in Subsection 5.5.2.

| | size | | | objective value | | | | sec | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| no. | $n$ | $k$ | $l$ | $\mu_{\text{mom}}(2)$ | $\mu_{\text{mom}}(4)$ | $\mu_{\text{sos}}(2)$ | $\mu_{\text{sos}}(4)$ | $\mu_{\text{mom}}(2)$ | $\mu_{\text{mom}}(4)$ | $\mu_{\text{sos}}(2)$ | $\mu_{\text{sos}}(4)$ |
| 13 | 2 | 4 | 6 | -0.698 | -0.698 | -0.698 | -0.698 | 0.34 | 29.64 | 0.53 | 30.24 |
| 14 | 2 | 6 | 6 | -0.037 | -0.037 | -0.037 | -0.037 | 0.41 | 39.04 | 0.58 | 39.15 |
| 15 | 2 | 8 | 6 | 0.787 | 0.787 | 0.787 | 0.787 | 0.49 | 38.69 | 0.64 | 59.17 |
| 16 | 2 | 10 | 6 | 0.015 | 0.015 | 0.015 | 0.015 | 0.74 | 87.78 | 0.79 | 98.50 |
| 17 | 3 | 4 | 6 | -1.902 | -1.902 | -1.902 | -1.902 | 0.58 | 79.53 | 0.73 | 83.27 |
| 18 | 3 | 6 | 6 | -0.317 | -0.317 | -0.317 | -0.317 | 0.52 | 84.51 | 0.80 | 122.25 |
| 19 | 3 | 8 | 6 | 0.196 | 0.196 | 0.196 | 0.196 | 1.17 | 121.75 | 0.93 | 138.92 |
| 20 | 3 | 10 | 6 | -0.668 | -0.668 | -0.668 | -0.668 | 1.31 | 252.30 | 1.13 | 323.83 |

Table 5.5: Computational test of containment of randomly generated spectrahedra as described in Example 5.17.

### 5.5.1 $S_A$ is a Polytope

In the special situation that $S_A = P_A$ is a polytope, we can make a more refined statement. We can solve a variation of the polynomial optimization problem (5.2) by applying the branch and bound scheme developed in the last chapter.

Assume problem (5.2) has a negative optimal value, i.e., $\mu < 0$ at some optimal point. By the same argument as in the proof of Proposition 5.5, there is another solution with negative objective value such that the $z$-component of the solution lies in the unit cube. This means that the following optimization problem also has a negative optimal value.

$$\begin{aligned} \text{minimize} \quad & z^T B(x) z \\ \text{s.t.} \quad & x \in P_A \\ & z \in [-1,1]^l \end{aligned} \tag{5.6}$$

On the other hand, if we have containment, the optimal value of (5.6) is equal to zero. Then the optimal point has a $z$-component that is zero. Problem (5.6) is a polynomial optimization problem over a polytope. Thus we can solve it by means of the branch and bound scheme from the last chapter.

Note that we do not have to solve to optimality, since we are only interested in detecting whether the optimal value is negative or non-negative. That means that we can add the following additional stopping criteria.

- STOP if the global lower bound is non-negative, i.e., $p_{lb,K} \geq 0$,
- STOP if any feasible point evaluates negative, i.e., $p_{ub,K} < 0$.

If the second criterion is met, the objective function is evaluated negative on some point $(x, z) \in P_A \times [-1,1]^l$, which was generated by the Deletion-by-Infeasibility step. This point serves as a certificate of non-containment, since for the $x$-component, we have $x \in P_A \setminus S_B$.

### 5.5.2 $S_A$ is a Spectrahedron

In the spirit of Section 4.3, it is of course possible to apply the branch and bound method to more general Positivstellensatz relaxations. While we do not have a theoretical convergence result at hand, from a geometric point of view this approach is also reasonable for the containment hierarchy.

Branching in the $x$-variables is reasonable because if $S_A$ is contained in $S_B$, the same must be true for any partition of $S_A$. Furthermore, we can restrict the branching to the $x$-variables. Overshooting the box for the $z$-coordinates may change the objective value of the relaxation but does not change the sign and does not influence the containment decision.

The challenges in this case are again to prove convergence and to find a suitable Deletion by Infeasibility method. For the convergence, explicit error bounds are needed which are not in sight.

For the Deletion by Infeasibility method, an algorithmically inexpensive test is to evaluate the pencil on the vertices of the partition polytopes and to test if the resulting scalar matrix is positive semidefinite. In this way, we can detect situations where one of the vertices lies in the spectrahedron. Of course, if all vertices evaluate to matrices that are not positive semidefinite, we cannot deduce infeasibility and have to retreat to another method.

In this case, we have to solve semidefinite feasibility problems. Recall that a linear pencil $A(x)$ is called strongly infeasible if $\operatorname{dist}(\{A(x) \mid x \in \mathbb{R}^n\}, \mathcal{S}_k^+) > 0$ and weakly infeasible if it is infeasible but not strongly infeasible. Detecting strongly infeasible pencils is not hard, in this situation we have a result similar to the Farkas' Lemma in the linear setting.

**Lemma 5.18** ([86]). *If a linear pencil $A(x)$ is strongly infeasible then*

$$-1 \in \left\{ s + \langle A, S \rangle \mid s \in \mathbb{R}_+,\ S \in \mathcal{S}_k^+ \right\}.$$

Detecting weakly infeasible problems requires, possibly expensive, reformulations of the original problem. See [57] for recent results.

In Section 4.2.1, we saw that we do not necessarily need a Deletion by Infeasibility method. Infeasible sets can also be detected by the bounding method, because they have an unbounded objective value, since in this case, -1 is a positive combination of the constraints. We have seen that the same is true in the spectrahedral setting, if the pencil is strongly feasible. For weakly feasible pencils, a similar result is also available.

**Theorem 5.19** (Klep, Schweighofer [42, Theorem 4.3.3.]). *Let $S_A = \emptyset$ and $t = \min(k-1, n)$. Then*

$$-1 \in \left\{ s + \langle A, S \rangle \mid s \in \Sigma_{n,2^t},\ S \in \Sigma_{n,2^t}^{k \times k} \right\}.$$

Notice however, that the degree of the certificate may need to be larger than the relaxation order used by default in the relaxation, which may make the bounding step very expensive.

Most modern semidefinite programming solvers can detect infeasibility. For our computations below we used the routine provided by MOSEK.

To test the ideas described above, we implemented the following branch and bound scheme.

**Containment branch and bound**

**Selection:** We implemented two different Selection rules:

Breadth First: Select the element $H_j$ with the greatest longest edge from $\mathcal{K}_{j-1}$.

Best Bound: Select the element $H_j$ with the lowest lower bound $p_{lb,H}$.

**Subdivision:** Subdivide $H_j$ by bisecting along the longest edge into $H_{j_1}$ and $H_{j_2}$.

**Deletion by Infeasibility:** Use the routine provided by MOSEK to decide if $H_{j_i} \cap P = \emptyset$, and if nonempty, generate a point $x$ in $H_{j_i} \cap S_A$.

**Bounding:** Calculate the lower bound on $H_{j_i} \cap S_A$ using the initial relaxation of the containment hierarchy (5.3). Calculate the upper bound by evaluating the pencil $B(x)$ on the feasible point $x$ returned by the linear program from the Deletion by Infeasibility step, use the smallest eigenvalue of this scalar matrix as upper bound.

**Stopping Criteria:**

STOP if the global lower bound is non-negative, i.e., $p_{lb,K} \geq 0$,

STOP if any feasible point evaluates negative, i.e., $p_{ub,K} < 0$.

As discussed in Section 4.3, both selection rules are bound improving and guarantee the convergence of the method. For our application, the best bound rule exhibits a better performance as we will see in the next example. We are not interested in solving the problem to optimality but rather in finding a certificate for non-containment quickly. By constantly improving the lower bound, we can find a positive bound early on in the case of containment. If we do not have containment, the partition set with the lowest bound is also a good candidate for a set that contains a point with negative objective value. This also explains the smaller number of bisections needed in the best bound approach in the following example.

**Example 5.20.** We applied the Containment branch and bound scheme described above to the problems from Table 5.4 and Table 5.5. The number of branching operations needed, depending on the selection rule, are reported in Table 5.6 in the column "steps", the column "cputime" gives the cputime in seconds needed to determine the status of the containment problem. It becomes apparent that the breadth first selection is indeed outperformed in most cases by the best bound rule.

To illustrate the approach, we end this chapter with Figure 5.3. It shows again the spectrahedra from experiment no. 4. The point marked by an asterisk is the point returned by the branch and bound scheme above as a certificate of non-containment.

| no. | status | selection rule: breadth first steps | breadth first cputime(s) | best bound steps | best bound cputime(s) |
|---|---|---|---|---|---|
| 1 | certified containment | 0 | 1.17 | 0 | 1.22 |
| 2 | certified containment | 0 | 1.69 | 0 | 1.69 |
| 3 | certified non-containment | 2 | 7.67 | 2 | 7.37 |
| 4 | certified non-containment | 4 | 12.89 | 3 | 9.75 |
| 5 | certified non-containment | 1 | 4.85 | 1 | 4.73 |
| 6 | certified non-containment | 20 | 108.94 | 5 | 33.35 |
| 7 | certified containment | 0 | 1.44 | 0 | 1.35 |
| 8 | certified containment | 0 | 2.16 | 0 | 2.15 |
| 9 | certified containment | 0 | 2.92 | 0 | 2.97 |
| 10 | certified non-containment | 1 | 5.76 | 1 | 5.57 |
| 11 | certified containment | 0 | 3.13 | 0 | 3.04 |
| 12 | certified non-containment | 38 | 292.57 | 9 | 70.11 |
| 13 | certified non-containment | 4 | 11.86 | 3 | 9.08 |
| 14 | certified non-containment | 3 | 13.99 | 3 | 13.37 |
| 15 | certified non-containment | 1 | 7.32 | 1 | 7.08 |
| 16 | certified non-containment | 29 | 101.83 | 10 | 37.54 |
| 17 | certified non-containment | 49 | 263.26 | 8 | 43.02 |
| 18 | certified non-containment | 54 | 403.24 | 9 | 74.41 |
| 19 | certified containment | 0 | 2.54 | 0 | 2.46 |
| 20 | certified containment | 0 | 3.73 | 0 | 3.95 |
| 21 | certified containment | 0 | 6.03 | 0 | 6.02 |
| 22 | certified containment | 0 | 3.19 | 0 | 3.17 |
| 23 | certified non-containment | 9 | 90.1 | 5 | 54.98 |
| 24 | certified containment | 0 | 8.09 | 0 | 7.98 |

Table 5.6: Number of bisections and cputime needed to certify containment or non-containment in the containment branch and bound scheme depending on the used selection rule as described in Example 5.20.

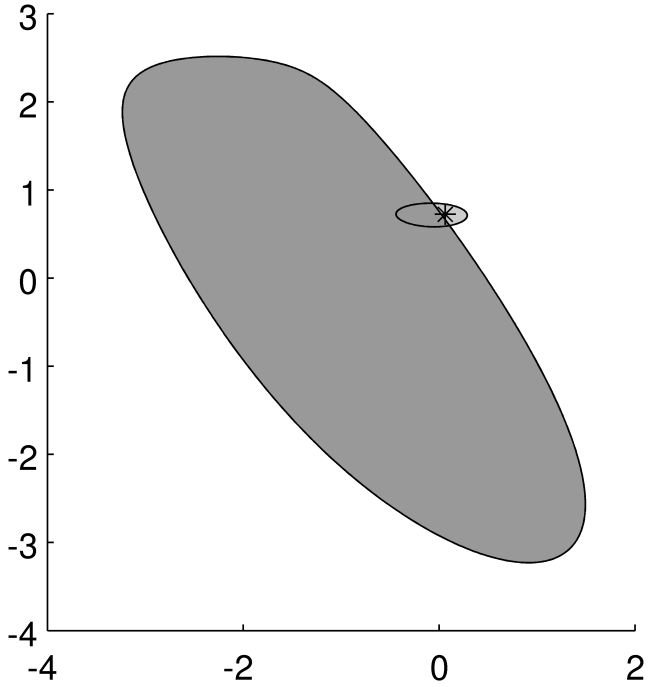


Figure 5.3: Spectrahedra from experiment no. 4 with point used as certificate of infeasibility (indicated by star). $S_A$: light grey, $S_B$: dark grey.

# 6 Open Problems

This chapter is meant as an entry point to topics and problems, which are closely related to this thesis, but were not in the scope of the present work. We will mainly list some open problems that we regard worth exploring.

In this work, we explored the combination of Positivstellensatz relaxations for polynomial programming problems with branch and bound techniques. Our main contribution here is the proof of convergence of a method based on Handelman's Positivstellensatz. The key ingredient for this result was the error bound derived in Chapter 3. It gives a bound on the error that is made when solving the linear programming relaxation based on Handelman's result of a polynomial programming problem over a polytope. The bound depends on the length of the longest edge of a bounding box and can thus be used to bound the error in the branching scheme.

We implemented a basic version of the proposed Handelman branch and bound method. It proved to work well on the elementary test problems studied in Chapter 4. Our implementation was not optimized for speed (setting up the linear programs sometimes takes longer than solving them) and does not compete with state of the art nonlinear programming codes. However, we have seen that strong bounding methods coming from Positivstellensatz relaxations lead to quickly converging methods. Our first question is concerned with the potential of such methods.

**Question 6.1.** What is the additional potential of refined implementations, and how do they perform on practical problems? How do refined implementations compare with other state of the art software?

In practical applications, the most commonly used Positivstellensatz is Putinar's. This is due to the fact that the number of terms in the representation stays relatively small (compared to Schmüdgen's and Handelman's Positivstellensatz) for increasing orders, while still reasonably good representation results are achieved. The drawback is that the sum of squares multipliers result in semidefinite instead of linear programming problems.

We expect that a convergence result similar to what we showed for Handelman's approach, can be devised for a branch and bound method using Putinar's Positivstellensatz for the lower bounding. At present, the following two approaches to proving convergence of the Putinar approach look most promising to us.

1) Generalizing the bound given by Nie and Schweighofer (Theorem 3.3). In their paper [69], they describe how the bound can be adapted to the situation, where the semialgebraic set $K$ is contained in a hypercube of edge length $r$. By thoroughly

reviewing the proof, this bound could even be made explicit for a simple family of sets (such as hyperrectangular sets), which in turn could be used as partitioning sets in a branch and bound scheme.

2) Generalizing the approach described by Magron [60] from hypercubes to hyperrectangular boxes. He describes how to transfer existing bounds for the Handelman approach to Putinar's approach under the assumption that the following conjecture by De Klerk and Laurent is true.

**Conjecture 6.2** ([13])**.**

$$\prod_{i=1}^{n} x_i + C_n \in M(G),$$

where $M(G)$ is the quadratic module spanned by the polynomials

$$G = \{x_1 - x_1^2, \ldots, x_n - x_n^2\}$$

describing the hypercube and

$$C_n = \frac{1}{n(n+2)}.$$

Strongly believing in the validity of the Conjecture, we ask the following.

**Question 6.3.** Can the error term $C_n$ in Conjecture 6.2 be made dependent on the size of the box to generalize Magron's result to arbitrary boxes?

The second area in which this work made a contribution is the study of spectrahedral containment problems. We extended complexity results from the polyhedral setting to the spectrahedral setting. The next generalization in this direction is to consider projections of spectrahedra (spectrahedral shadows). The positive results, that is "$\mathcal{V}$ in $\mathcal{S}$" and "$\mathcal{H}$ in $\mathcal{S}$" from Table 5.1 seem to be easily transferred to this new setting. The study of the co-NP-hard cases is more involved.

**Question 6.4.** Can the relaxation techniques for containment of spectrahedra introduced in this work be generalized to projections of spectrahedra?

With the hierarchy (5.3) we found a sufficient criterion for the containment question "$\mathcal{S}$ in $\mathcal{S}$" and attacked one of the two main cases classified as co-NP-hard in Table 5.1. The next questions concerns the other hardness result.

**Question 6.5.** Can we find good relaxations for the containment question "$\mathcal{H}$ in $\mathcal{V}$" that can be decided efficiently?

The sufficient containment criteria (5.4) and (5.3) can be interpreted in terms of maps between the matrix (sub-)spaces spanned by the coefficient matrices of the pencils $A(x)$ and $B(x)$. Criterion (5.4) is related to completely positive maps (see [28, 40, 41]), the hierarchy (5.3) to positive maps (see [41]). Such maps are usually studied in the setting of operator algebras. We believe it to be fruitful to further study the connection between positivity in operator theory and containment questions.

# Bibliography


[1] M. Abril Bucero and B. Mourrain. Exact relaxation for polynomial optimization on semi-algebraic sets. *ArXiv: 1307.6426v4*, July 2013.

[2] E.D. Andersen and K.D. Andersen. Presolving in linear programming. *Math. Program.*, 71(2, Ser. A):221–245, 1995.

[3] E.D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Math. Program.*, 95(2, Ser. B):249–277, 2003.

[4] M.F. Anjos and J.B. Lasserre, editors. *Handbook on semidefinite, conic and polynomial optimization.* International Series in Operations Research and Management Science, 166. Springer, New York, 2012.

[5] Gennadiy Averkov. Constructive proofs of some Positivstellensätze for compact semialgebraic subsets of $\mathbb{R}^d$. *J. Optim. Theory Appl.*, 158(2):410–418, 2013.

[6] A. Ben-Tal and A. Nemirovski. On tractable approximations of uncertain linear matrix inequalities affected by interval uncertainty. *SIAM J. Optim.*, 12(3):811–833, 2002.

[7] S.N. Bernstein. Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. *Comm. Soc. Math. Kharkow*, 13:1–2, 1912.

[8] G. Blekherman, P.A. Parrilo, and R.R. Thomas. *Semidefinite Optimization and Convex Algebraic Geometry.* SIAM, Philadelphia, PA, 2013.

[9] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, Cambridge, 2004.

[10] S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing - STOC '71*, pages 151–158, New York, USA, 1971. ACM Press.

[11] R.J. Dakin. A tree-search algorithm for mixed integer programming problems. *Comput. J.*, 8:250–255, 1965.

[12] E. de Klerk. *Aspects of semidefinite programming*, volume 65 of *Applied Optimization.* Kluwer Academic Publishers, Dordrecht, 2002.

[13] E. de Klerk and M. Laurent. Error bounds for some semidefinite programming approaches to polynomial minimization on the hypercube. *SIAM J. Optim.*, 20(6):3104–3120, 2010.

[14] E. de Klerk and M. Laurent. On the Lasserre hierarchy of semidefinite programming relaxations of convex polynomial optimization problems. *SIAM J. Optim.*, 21(3):824–832, 2011.

[15] Q. Feng, H. Mao, J. Wang, and H. Jiao. A branch and bound algorithm for generalized polynomial programming problems. In *International Conference on Information Engineering and Computer Science (ICIECS)*, pages 1–4, 2009.

[16] R.M. Freund and J.B. Orlin. On the complexity of four polyhedral set containment problems. *Math. Program.*, 33(2):139–145, 1985.

[17] K. Fukuda. Frequently Asked Questions in Polyhedral Computation . `http://www.cs.mcgill.ca/~fukuda/soft/polyfaq/polyfaq.html`. Online; accessed: June 22, 2014.

[18] J. Garloff. Convergent bounds for the range of multivariate polynomials. In *Interval mathematics, 1985 (Freiburg, 1985)*, volume 212 of *Lecture Notes in Comput. Sci.*, pages 37–56. Springer, Berlin, 1986.

[19] B. Ghaddar, J. C. Vera, and M. F. Anjos. An iterative scheme for valid polynomial inequality generation in binary polynomial programming. In *Integer programming and combinatorial optimization*, volume 6655 of *Lecture Notes in Comput. Sci.*, pages 207–222. Springer, Heidelberg, 2011.

[20] A.J. Goldman and M.V. Ramana. Some geometric results in semidefinite programming. *J. Global Optimization*, 7(1):33–50, 1995.

[21] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[22] J. Gouveia and T. Netzer. Positive polynomials and projections of spectrahedra. *SIAM J. Optim.*, 21(3):960–976, 2011.

[23] P. Gritzmann and V. Klee. Inner and outer $j$-radii of convex bodies in finite-dimensional normed spaces. *Discrete Comput. Geom.*, 7(3):255–280, 1992.

[24] P. Gritzmann and V. Klee. Computational complexity of inner and outer $j$-radii of polytopes in finite-dimensional normed spaces. *Math. Program.*, 59(2, Ser. A):163–213, 1993.

[25] P. Gritzmann and V. Klee. On the complexity of some basic problems in computational convexity. I. Containment problems. *Discrete Math.*, 136(1-3):129–174, 1994.

[26] D. Handelman. Representing polynomials by positive linear functions on compact convex polyhedra. *Pacific Journal of Mathematics*, 132(1):35–62, 1988.

[27] G.H. Hardy, J.E. Littlewood, and G. Pólya. *Inequalities*. Cambridge, at the University Press, 1934.

[28] J.W. Helton, I. Klep, and S. McCullough. The matricial relaxation of a linear matrix inequality. *Math. Program.*, 138(1-2, Ser. A):401–445, 2013.

[29] J.W. Helton and V. Vinnikov. Linear matrix inequality representation of sets. *Comm. Pure Appl. Math.*, 60(5):654–674, 2007.

[30] D. Henrion and J.B. Lasserre. Detecting global optimality and extracting solutions in gloptipoly. In Didier Henrion and Andrea Garulli, editors, *Positive Polynomials in Control*, volume 312 of *Lecture Notes in Control and Information Science*, pages 293–310. Springer, Berlin, 2005.

[31] D. Henrion and J.B. Lasserre. Convergent relaxations of polynomial matrix inequalities and static output feedback. *IEEE Trans. Autom. Control*, 51(2):192–202, 2006.

[32] D. Hilbert. Über die Darstellung definiter Formen als Summe von Formenquadraten. *Math. Ann.*, 32(3):342–350, 1888.

[33] C.W.J. Hol and C.W. Scherer. Sum of squares relaxations for polynomial semidefinite programming. In *Proc. Symp. Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

[34] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1994.

[35] R. Horst. Deterministic global optimization with partition sets whose feasibility is not known: application to concave minimization, reverse convex constraints, DC-programming, and Lipschitzian optimization. *J. Optim. Theory Appl.*, 58(1):11–37, 1988.

[36] R. Horst and H. Tuy. *Global optimization. Deterministic approaches*. Springer, Berlin, third edition, 1993.

[37] T. Ibaraki. Theoretical comparisons of search strategies in branch-and-bound algorithms. *Internat. J. Comput. Information Sci.*, 5(4):315–344, 1976.

[38] M. Joswig and T. Theobald. *Algorithmische Geometrie*. Vieweg Studium: Aufbaukurs Mathematik. Vieweg, Wiesbaden, 2008.

[39] M.H. Karwan, V. Lotfi, J. Telgen, and S. Zionts. *Redundancy in Mathematical Programming*, volume 206 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin, 1983. A state-of-the-art survey, With contributions by Arnon Boneh, Gordon H. Bradley, Gerald G. Brown, Tomáš

Gál, Glenn W. Graves, S. Holm, Dieter Klein, Theodore H. Mattheiss, David S. Rubin, Awanti P. Sethi, Gerald L. Thompson, Jyrki Wallenius and H. Paul Williams.

[40] K. Kellner, T. Theobald, and C. Trabandt. Containment problems for polytopes and spectrahedra. *SIAM J. Optim.*, 23(2):1000–1020, 2013.

[41] K. Kellner, T. Theobald, and C. Trabandt. A semidefinite hierarchy for containment of spectrahedra. *ArXiv: 1308.5076v2*, 2013.

[42] I. Klep and M. Schweighofer. An exact duality theory for semidefinite programming based on sums of squares. *Math. Oper. Res.*, 38(3):569–590, 2013.

[43] M. Kojima. Sums of squares relaxations of polynomial semidefinite programs. Technical report, Research Report B-397, Dept. Math. Comput. Sc., Tokyo Inst. Tech, Japan, 2003.

[44] A.H. Land and A.G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[45] J.M. Lane and R.F. Riesenfeld. Bounds on a polynomial. *BIT*, 21(1):112–117, 1981.

[46] J.B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796, 2001.

[47] J.B. Lasserre. Polynomial programming: LP-relaxations also converge. *SIAM J. Optim.*, 15(2):383–393, 2004.

[48] J.B. Lasserre. A unified criterion for positive definiteness and semidefiniteness. *Research Report 05283, LAAS CNRS Roulouse, France*, 2005.

[49] J.B. Lasserre. Convergent SDP relaxations in polynomial optimization with sparsity. *SIAM J. Optim.*, 17(3):822–843, 2006.

[50] J.B. Lasserre. A sum of squares approximation of nonnegative polynomials. *SIAM Rev.*, 49(4):651–669, 2007.

[51] J.B. Lasserre. *Moments, Positive Polynomials and Their Applications*. Imperial College Press optimization series. Imperial College Press, 2010.

[52] J.B. Lasserre. A lagrangian relaxation view of linear and semidefinite hierarchies. *SIAM J. Optim.*, 23(3):1742–1756, 2013.

[53] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In Mihai Putinar and Seth Sullivant, editors, *Emerging Applications of Algebraic Geometry*, volume 149 of *The IMA Volumes in Mathematics and its Applications*, pages 157–270. Springer, New York, 2009.

[54] E.L. Lawler and D.E. Wood. Branch-and-bound methods: A survey. *Operations Res.*, 14(4):699–719, 1966.

[55] R. Leroy. *Certificats de positivité et minimisation polynomiale dans la base de Bernstein multivariée*. PhD thesis, IRMAR - Institut de Recherche Mathématique de Rennes, 2008.

[56] R. Leroy. Certificates of positivity in the simplicial Bernstein basis. `http://hal.archives-ouvertes.fr/hal-00589945`, 2011.

[57] M. Liu and G. Pataki. Exact duality in semidefinite programming based on elementary reformulations. *ArXiv: 1406.7274*, June 2014.

[58] J. Löfberg. Yalmip: A toolbox for modeling and optimization in MATLAB. In *Proc. CACSD Conference*, pages 284–289, Taipei, Taiwan, 2004.

[59] J. Löfberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Trans. Autom. Control*, 54(5):1007–1011, 2009.

[60] V. Magron. Error bounds for polynomial optimization over the hypercube using Putinar type representations. *Optimization Letters*, 8(7):1–9, 2014.

[61] S. Malan, M. Milanese, M. Taragna, and J. Garloff. B 3 algorithm for robust performances analysis in presence of mixed parametric and dynamic perturbations. In *Proceedings of the 31st IEEE Conference on Decision and Control, 1992*, pages 128–133. IEEE, 1992.

[62] O.L. Mangasarian. Set containment characterization. *J. Global Optim.*, 24(4):473–480, 2002.

[63] P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.

[64] H.D. Mittelmann. Benchmarks for optimization software. `http://plato.la.asu.edu/bench.html`. Online; accessed: June 13, 2014.

[65] P.S.V. Nataraj and M. Arounassalame. Constrained global optimization of multivariate polynomials using Bernstein branch and prune algorithm. *J. Global Optim.*, 49(2):185–212, 2011.

[66] J. Nie. An exact Jacobian SDP relaxation for polynomial optimization. *Math. Program.*, 137(1-2, Ser. A):225–255, 2013.

[67] J. Nie. Optimality Conditions and Finite Convergence of Lasserre's Hierarchy. *Math Program.*, (Ser. A), to appear.

[68] J. Nie, J. Demmel, and B. Sturmfels. Minimizing polynomials via sum of squares over the gradient ideal. *Math. Program.*, 106(3, Ser. A):587–606, 2006.

[69] J. Nie and M. Schweighofer. On the complexity of Putinar's Positivstellensatz. *J. Complexity*, 23(1):135–150, 2007.

[70] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, California Institute of Technology, 2000.

[71] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming B*, 96(2):293–320, 2003.

[72] G. Pólya. Über positive Darstellung von Polynomen. *Vierteljschr. Naturforsch. Ges. Zürich*, 73:141–145, 1928.

[73] V. Powers and B. Reznick. A new bound for Pólya's theorem with applications to polynomials positive on polyhedra. *J. Pure Appl. Algebra*, 164(1-2):221–229, 2001.

[74] A. Prestel and C.N. Delzell. *Positive polynomials.* Springer Monographs in Mathematics. Springer, Berlin, 2001.

[75] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[76] M.V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Math. Program.*, 77(1):129–162, 1997.

[77] T.J. Rivlin. Bounds on a polynomial. *J. Res. Nat. Bur. Standards Sect. B*, 74B:47–54, 1970.

[78] J. Rokne. Bounds for an interval polynomial. *Computing*, 18(3):225–240, 1977.

[79] J. Rokne. Optimal computation of the Bernstein algorithm for the bound of an interval polynomial. *Computing*, 28(3):239–246, 1982.

[80] C.W. Scherer and C.W.J. Hol. Matrix sum-of-squares relaxations for robust semi-definite programs. *Math. Program.*, 107(1-2):189–211, 2006.

[81] K. Schmüdgen. The $K$-moment problem for compact semi-algebraic sets. *Math. Ann.*, 289(2):203–206, 1991.

[82] D. Scholz. *Deterministic global optimization*, volume 63 of *Springer Optimization and Its Applications.* Springer, New York, 2012.

[83] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Ltd., Chichester, 1986.

[84] M. Schweighofer. On the complexity of Schmüdgen's positivstellensatz. *J. Complexity*, 20(4):529–543, 2004.

[85] A.P. Smith. *Enclosure Methods for Systems of Polynomial Equations and Inequalities*. PhD thesis, Universität Konstanz, 2012.

[86] J. F. Sturm. Theory and algorithms of semidefinite programming. In *High performance optimization*, volume 33 of *Appl. Optim.*, pages 1–194. Kluwer Acad. Publ., Dordrecht, 2000.

[87] B.Y. Su, B. Catanzaro, and N. Sundaram. Branch & bound pattern. `http://parlab.eecs.berkeley.edu/wiki/patterns/backtrack_branch_and_bound`. Online; accessed: May 7, 2014.

[88] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

# Curriculum Vitæ

## Christian Wilhelm Trabandt


| | |
|---|---|
| Address: | Landgrafenstr. 33,<br>60486 Frankfurt am Main |
| Phone: | 069-17523224 |
| E-Mail: | trabandt@math.uni-frankfurt.de |
| Date of Birth: | August 16th, 1985 |
| Place of Birth: | Bad Nauheim |


### Education

| | |
|---|---|
| April 2011 -<br>July 2014 | Goethe University, Frankfurt, Germany:<br>PhD Student in Mathematics,<br>Advisor: Thorsten Theobald |
| April 2006 -<br>October 2010 | Goethe University, Frankfurt, Germany:<br>Mathematics Diplom, Grade: "very good"<br>Title of thesis: "LMI-Representable Sets"<br>Advisor: Thorsten Theobald |
| 1998 - 2005 | St. Lioba Schule, Bad Nauheim, Germany:<br>Abitur (university entrance diploma) |

### Honors and Awards

| | |
|---|---|
| June 2011 -<br>May 2014 | Studienstiftung des deutschen Volkes<br>(German National Scholarship Foundation),<br>Bonn, Germany:<br>Scholarship (full financial support) |
| July 2011 | Goethe University, Frankfurt Germany:<br>Award for best Mathematics Diplom in 2010 |

### Employment History:

| | |
|---|---|
| April 2011 –<br>June 2011 | Goethe University, Frankfurt, Germany:<br>Research Assistant |
| September 2007<br>- February 2009 | Goethe University, Frankfurt, Germany:<br>Teaching Assistant |

## Service:

| | |
|---|---|
| 2012 - 2013 | MEGA 2013 (Enumerative Methods in Algebraic Geometry), Frankfurt, Germany:<br>Member of the local committee |
| 2008 | Goethe University, Frankfurt, Germany:<br>Member of the appointment committees "Partial Differential Equations" and "Differential Geometry" |

## Publications:

| | |
|---|---|
| May 2014 | K. Kellner, T. Theobald, and C. Trabandt. "A Semidefinite Hierarchy for Containment of Spectrahedra", *to appear in SIAM Journal on Optimization.* |
| May 2013 | K. Kellner, T. Theobald, and C. Trabandt. "Containment problems for polytopes and spectrahedra", *SIAM Journal on Optimization* 2013, Vol. 23(2), 1000-1020. |