

Open Journal of Mathematical Optimization

Gennadiy Averkov, Benjamin Peters & Sebastian Sager

Unifying view on sparse convex relaxations in polynomial optimization

Volume 7 (2026), article no. 2 (44 pages)

<https://doi.org/10.5802/ojmo.50>

Article submitted on March 3, 2025, revised on January 22, 2026,
accepted on April 25, 2026.

© The author(s), 2026.



This article is licensed under the
CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.
<http://creativecommons.org/licenses/by/4.0/>



Unifying view on sparse convex relaxations in polynomial optimization

Gennadiy Averkov

Fakultät 1, Brandenburgische Technische Universität Cottbus-Senftenberg, Germany
averkov@b-tu.de

Benjamin Peters

Otto-von-Guericke Universität Magdeburg, Germany
benjamin@benjamin-peters.de

Sebastian Sager

Otto-von-Guericke Universität Magdeburg, Germany
Max Planck Institute for the Dynamics of Complex Technical Systems Magdeburg, Germany
sager@ovgu.de

Abstract

We present a novel, general, and unifying point of view and use it to surveys sparse and dual approaches to polynomial optimization. Solving polynomial optimization problems to global optimality is a ubiquitous challenge in many areas of science and engineering. Different approaches on how to solve nonconvex polynomial optimization problems based on convex relaxations have been developed in different scientific communities. Here, we introduce the concept of monomial patterns. A pattern determines what monomials are to be linked by convex constraints in a convex relaxation of a polynomial optimization problem. This concept helps understanding existing approaches from different schools of thought, developing novel relaxation schemes, and deriving a flexible duality theory, which can be specialized to many concrete situations that have been considered in the literature. We survey different approaches to polynomial optimization including polyhedral approximations, dense semidefinite relaxations, SONC, SAGE, and TSSOS in a self-contained, unifying exposition.

Digital Object Identifier 10.5802/ojmo.50

Keywords Polynomial optimization, nonlinear optimization, convexification, sparsity, duality, sum of squares, SONC, SAGE, TSSOS, SDSOS.

Acknowledgments This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 314838170, GRK 2297 MathCoRe.

1 Introduction

Let X be a basic closed semi-algebraic set defined as

$$X = \{x \in \mathbb{R}^n : g_1(x) \geq 0, \dots, g_s(x) \geq 0, h_1(x) = 0, \dots, h_t(x) = 0\} \quad (1)$$

via s polynomial inequality and t polynomial equality constraints, where $s, t \geq 0$ are non-negative integers. In polynomial optimization, we are interested in minimizing a polynomial over the *feasible set* X . We want to solve the problem

$$\inf_{x \in X} f(x), \quad (\text{POP})$$

where the objective f and the polynomials used to describe X are polynomials, i.e., elements of the ring $\mathbb{R}[x]$ of n -variate polynomials in the variables $x = (x_1, \dots, x_n)$ with coefficients in \mathbb{R} . We use $x^\alpha := x_1^{\alpha_1} \dots x_n^{\alpha_n}$ for the monomials in x , where a monomial is determined by its exponent vector $\alpha \in \mathbb{N}^n$. In this notation, $f(x) = \sum_{\alpha \in A} f_\alpha x^\alpha \in \mathbb{R}[x]_A$ is an \mathbb{R} -linear combination of monomials, where A is a finite set of exponent vectors. Where convenient, we also use f to denote the vector of coefficients of $f(x)$. A list of symbols is provided for convenience in Appendix A.



© Gennadiy Averkov & Benjamin Peters & Sebastian Sager;
licensed under Creative Commons License Attribution 4.0 International

1.1 The problem class polynomial optimization

We emphasize that certain important aspects of polynomial optimization are out of scope of this article. Among them are infinite dimensions, i.e., the limit case of the approximation hierarchies discussed below [38], the important role of Newton polytopes for exploiting sparsity [57, 79], symmetry exploitation [28, 31, 51, 60], hybrid relaxation strategies [7], and scalable algorithms [84]. We see these aspects as complementary to our tutorial-style generalization of polynomial optimization concepts and refer to the cited papers, respectively, for detailed discussions and further references.

To get a first intuition for the problem class, we discuss polynomials and optimization from a general perspective, before we look into aspects of how to solve (POP).

1.1.1 Polynomials and Motivating Applications

In polynomial optimization, the objective function and the inequalities and equalities describing X are polynomials in real variables $x \in \mathbb{R}^n$. They form a huge and important subset in the more general class of nonlinear programming (NLP) problems.

Polynomial optimization problems of type (POP) appear in a variety of different areas and contexts, due to the ubiquity of polynomials and their properties. Polynomials arise directly in the mathematical modeling of real world systems, e.g., with a cubic function in the Fitzhugh–Nagumo model or with a quadratic function when air friction is modeled. They also arise from algebraic concepts. A famous example is the Routh–Hurwitz stability criterium which states that a time-invariant linear system is stable if the roots of its characteristic polynomial have only negative real parts. Another example are symbolic reformulations and underestimators using polynomials, e.g., [67, 68]. Even binary variables $x_i \in \{0, 1\}$ can be modeled via quadratic constraints $x_i(1 - x_i) = 0$. The seminal work of Goemans and Williamson on deriving semidefinite programming based bounds for the combinatorial max cut and satisfiability problems [25] boosted interest in more general relaxations for (POP). For many open problems in mathematics and computer science such as contributing to proving or disproving Khot’s unique games conjecture [59] or gaining insight into the kissing number problem [48] polynomial representations and semidefinite programming approximations have been used successfully. Concrete examples of applications in polynomial optimization, e.g., in discrete and combinatorial optimization, control systems and robotics, statistics, and electric power systems engineering are provided in [3, Section 1.1], [39, Section 1.1], and [13, Section 3.6].

Yet another reason for the importance of polynomials is their universal approximation property. Weierstraß showed in 1885 that the subalgebra of polynomials is dense in the algebra of continuous functions (a property much celebrated and used nowadays for neural networks). The Taylor series of a function provides a convenient way to find polynomial approximations and error estimates for smooth functions. Polynomial regression and approximation are also computationally stable and efficient, e.g., using Tschebysheff polynomials and barycentric coefficients [76]. Therefore it is certainly no exaggeration to claim that mastering of the problem class (POP) would have a huge impact on many open challenges in science and engineering.

The structure of X and the way it is given is of crucial importance for particular approaches. For (POP), it is already interesting and nontrivial to address the special situations in which $X = \mathbb{R}^n$, which does not require any constraints, $X = [-1, 1]^n$, modeled by the linear inequalities $-1 \leq x_i \leq 1$, or $X = \{-1, 1\}^n$ modeled by the equalities $x_i(1 - x_i) = 0$. All those cases have a different flavor, and there are methods specific to those cases, which can be employed for solving a particular type of polynomial optimization problems. Nevertheless, independent of a particular version of (POP), convexification is a common principle applied to all these versions. In the following we shall explain this universal principle without going into details regarding X , unless otherwise specified.

1.1.2 Optimization

A feasible point x^* is called a global minimizer of (POP), if there is no other point in X with a lower objective function value. It is called a local minimizer, if there exists a neighborhood in X around x^* in which this property holds. It is way more difficult to find a global optimizer or even to verify global optimality of a given point. In contrast, local optimality can usually be verified locally via sufficient conditions of optimality by evaluating functions and higher-order derivatives in the candidate point.

Convex optimization, i.e., the minimization of a convex objective function over a convex feasible set X , has two important advantages compared to nonconvex optimization. The first advantage is that for convex functions,

the necessary first-order conditions of optimality are also sufficient conditions. The second, more important advantage is that a local minimizer of a convex problem is also a global minimizer. This can be seen by assuming the existence of two strict local minimizers, considering the connecting line of (feasible) points and constructing a contradiction to the local optimality of both from the convexity assumption on the objective function. For convex nonlinear problems, efficient algorithms have been developed in the last decades as shortly discussed in Section 1.4.

Many practical optimization problems are nonconvex, though. Sometimes, the determination of local optima is acceptable in an engineering context, based on the rationale that a descent-based algorithm initialized with the currently implemented practical solution will improve it towards a local optimum. Practitioners in industry might even argue that sometimes, moving from one local solution to a different one would imply additional undesirable costs. Examples are the necessary training of staff to control a complex process or related security issues. In addition, finding global optimizers is simply often not realistically possible. Already local optimizers can be very difficult to find due to high computational costs, e.g., in the training of deep neural networks.

In many other contexts, apart of the intellectual desire to know what the very best solution is, global optimality is required. This is certainly true when optimization results are used for mathematical reasoning, such as the determination of bounds for kissing numbers [48]. Note that here often additional concepts such as interval analysis are necessary to obtain rigorous results [52]. Global optimization comes in several flavors and versions and is supported and developed by a wide range of sub-communities representing different philosophies. In our review, we are interested in the global solution of (POP) and shall develop a general convexification framework for this purpose, which is also able to reduce computational costs via sparsification.

1.1.3 How to globally solve polynomial optimization problems

There are many excellent textbooks on polynomial optimization available [5, 13, 24, 37, 38, 45, 46, 53, 82] that highlight the connection to positive polynomials, sum of squares (SOS), semidefinite programming (SDP), and related concepts from real algebra. Important contributions come also from a second community interested in global nonlinear programming (NLP), [52]. Here, polyhedral relaxations and spatial branch and bound techniques [35, 41] have been suggested, e.g., [71, 73], sometimes in the connection with additional concepts such as interval arithmetics [49]. Often these approaches address also nonlinear optimal control problems, e.g., [1, 18, 20, 21, 55].

Many different approaches for the convexification of (POP) have been suggested in the literature. Among them are the well-known McCormick envelopes [14, 47, 64], i.e., the convexification of variables x_1 and x_2 and their product x_1x_2 . Other examples are truncated moment relaxation and its dual, the SOS relaxation, [5, 39, 46], scaled-diagonally-dominant sum of squares (SDSOS) [3], sums of non-negative circuit polynomials [19, 65], bound-factor products [16] and their dual Handelman's hierarchy [29], multilinear intermediates [10], polyhedral outer approximations [72] as well as expression trees [67, 68]. These approaches are either based on primal or on dual formulations, as shall become clearer in the course of this review. It is one goal of this paper to present a unifying point of view fostering a better understanding of the differences and similarities between these approaches. Also, a variety of optimization solvers has evolved over the years, as discussed in Appendix C.

Polynomial optimization problems have obvious connections with real algebra, which in turn, is connected to semidefinite optimization. Since the seminal work of Jean-Bernard Lasserre, in the last two decades a community of experts working at the interface of global optimization, semidefinite optimization, real algebra, and real algebraic geometry has formed. This community seeks to understand the theoretical foundations of convexification as *the* fundamental principle of global optimization and to use it in computations. It is interesting to observe the interplay between theory and computational practice in this context. The theory suggests tools of great generality supplied with extremely inefficient general algorithms. The computational practice exploits rather concrete convexifications (like McCormick envelopes), which can be deployed in general problems. Basically any kind of concretization of convexification ideas has led to a spin-off – a sub-culture focussing and propagating this particular convexification technique and its application for solving (POP). Examples are McCormick envelopes, other kinds of linear programming relaxations, Lasserre relaxations, SONC, and SAGE relaxations. With our review, we address three key similarities between these approaches from different sub-communities: first, using *convexification* as a main tool, second, trying to impose *sparse* constraints in practice, and third, establishing a *duality theory* based on general principles from conic optimization. In our opinion, these similarities have not yet been articulated clearly enough. In our presentation, we demonstrate how the idea of sparse convexifications is

implemented flexibly, without relying on specific modeling details. We illustrate the impact of convexification and sparsification concepts with numerical results.

1.2 Convexification

The (formal) convexification of problem (POP) consists of two elementary steps: lifting and describing the relationships between the variables x and v with convex constraints.

1.2.1 Monomial Lifting

By introducing a new variable $v_\alpha = x^\alpha$ for each of the monomials that occur in (POP), we linearize the objective, which becomes the linear function $(v_\alpha)_{\alpha \in A} \mapsto \sum_{\alpha \in A} f_\alpha v_\alpha$. We introduce the *linearization* map L_v for polynomials $f \in \mathbb{R}[x]$ by

$$L_v(f) := L_v\left(\sum_{\alpha \in A} f_\alpha x^\alpha\right) := \sum_{\alpha \in A} f_\alpha v_\alpha.$$

Since the monomial variable v_0 for the monomial $x^0 = 1$ is assigned the value 1, we use v_0 as the constant 1 if it is not specified otherwise. We call the linear inequality $L_v(f) \geq 0$ the linearization of the polynomial inequality $f \geq 0$. In [38, Section 2.7] L_v (without fixing $v_0 = 1$) is called the Riesz linearization functional. For example, the linearization of a polynomial $f := (1 - x_1)(1 - x_2)$ is given by $L_v(f) = L_v(1 - x_1 - x_2 + x_1x_2) = v_{0,0} - v_{1,0} - v_{0,1} + v_{1,1} = 1 - v_{1,0} - v_{0,1} + v_{1,1}$.

Having a linear objective is favorable, but it requires the introduction of nonlinear equality constraints $v_\alpha = x^\alpha$, with $\alpha \in A$, additionally to the constraint $x \in X$. This stage of processing (POP) is the lifting stage, as we add new variables and optimize in a higher-dimensional space. The variable v_α is called the monomial variable for the monomial x^α (or, for the exponent vector α). After this kind of reformulation, we have a vector $x = (x_1, \dots, x_n)$ that varies in X and determines the values of the monomial variables v_α , $\alpha \in A$, on which the objective $L_v(f)$ depends. Thus, we do not need x directly anymore, but we want to determine explicitly how the monomial variables, which are linked to each other through the x -variables, are interdependent.

1.2.2 Formal convexification

Problem (POP) is now reformulated as

$$\inf \left\{ \sum_{\alpha \in A} f_\alpha v_\alpha : x \in X, v_\alpha = x^\alpha (\alpha \in A) \right\}$$

The next step is to project out the x -variables, keeping the monomial variables and introducing constraints that describe their relations. Since we want to arrive at a convex problem in the end, we describe the relation in terms of convex constraints. This intention can be formalized by replacing the constraint

$$(v_\alpha)_{\alpha \in A} \in \{(x^\alpha)_{\alpha \in A} : x \in X\}, \tag{2}$$

which is in fact a family of the constraints $v_\alpha = x^\alpha$, indexed by $\alpha \in A$, with the convex constraint

$$(v_\alpha)_{\alpha \in A} \in \mathcal{M}_A(X) := \text{conv}\{(x^\alpha)_{\alpha \in A} : x \in X\}. \tag{3}$$

We have a linear objective on a convex feasible set, so it is readily clear that (POP) is equivalent to the problem

$$\inf \left\{ \sum_{\alpha \in A} f_\alpha v_\alpha : v \in \mathcal{M}_A(X) \right\} \tag{CVX-POP}$$

in the sense that (POP) and (CVX-POP) have the same optimal value. We emphasize that (CVX-POP), the convex problem which we call the monomial convexification of (POP), is a formal problem. By this we mean that (CVX-POP) provides no indication on how to formulate $\mathcal{M}_A(X)$ in any of the available optimization paradigms. Instead, (CVX-POP) simply declares our intention to reformulate or relax a nonconvex problem to a convex one. This process is called *convexification* in the jargon of global optimization. The set $\mathcal{M}_A(X)$ represents the (not yet known) convexification that conveys the (not yet known) convex constraints, through

which the monomial variables v_α , $\alpha \in A$, are related. Both in the theory of global optimization and real algebra, a major aim is to provide approaches to describe $\mathcal{M}_A(X)$, exactly or approximately, within tractable paradigms of convex optimization. Linear, second order-cone, and semidefinite optimization are the most widely used paradigms of convex optimization for that purpose. The points of view on what *tractable* means may vary among sub-communities. Theoretically inclined experts consider polynomial-time solvability of a class of optimization problems to be the standard definition of being tractable. In contrast, practitioners may apply a more refined distinction, as they also care about the conditioning of the problems, the rate of convergence and the exact order of polynomial running time. Hence, for example, in view of these remarks semidefinite optimization with a large size of semidefinite constraints can be considered to be tractable in the theoretically inclined community, but would be considered intractable in the practically inclined one.

1.3 Formal sparsification

We call a constraint *dense*, if it involves a large fraction of the variables of the underlying optimization problem. In contrast, a *sparse* constraint depends only on a small fraction of the optimization variables. More correctly, we are mainly interested in the difficulty of associated constraints and their impact on computational runtimes. E.g., when we use semidefinite modeling techniques, we measure the difficulty in terms of the sizes of the linear matrix inequalities, as discussed below in Section 1.4.

Whereas the question how to actually model the convex $\mathcal{M}_A(X)$ is definitely *the* core topic in global optimization, also the role of sparsity and approaches to impose sparsity are very important in practice. Suppose, for example, that we have developed an approach to model $\mathcal{M}_A(X)$ exactly or approximately with some theoretical guarantees. This means that we have established a method that transfers the formal constraint $v \in \mathcal{M}_A(X)$ into a set of concrete constraints,

$$\mathcal{M}_A(X) \xrightarrow{\text{modeling method}} \text{Model of } \mathcal{M}_A(X).$$

Our modeling method receives A and a description of X as an input and produces a model for $\mathcal{M}_A(X)$. But what if, when A is large and complicated, this model for $\mathcal{M}_A(X)$ is not tractable, because it involves computationally hard-to-handle or ill-conditioned constraints? At a first glance, our modeling method is not helpful: applying it to (CVX-POP) we obtain its concrete formulation, but we cannot solve it, because it is not tractable. But looking more precisely, we see that there is a remedy, and it relies on using sparse convex constraints. When we model $\mathcal{M}_A(X)$, we want to bind *all* of the variables v_α with $\alpha \in A$ by convex constraints. This is a tough task, because the number of variables may be large and their relations may be highly nontrivial. What we can do instead is creating smaller (and possible overlapping) groups of monomial variables and try to model the relations of variables within each single group. As one of the options, we can just use the *same* modeling method for each group of variables. The latter is a high-level view on imposing sparsity, which refrains from details on how to choose the sparse sub-structures and what kinds of sparse sub-structures are favorable. The above strategy leads to what we call a pattern relaxation of (POP) with respect to the pattern family $\{P_1, \dots, P_N\}$, which covers A :

$$\inf \left\{ \sum_{\alpha \in A} f_\alpha v_\alpha : (v_\alpha)_{\alpha \in P_i} \in \mathcal{M}_{P_i}(X) \text{ for } i \in [N] \right\} \quad (\text{P-RLX})$$

This is a formal problem that declares the intention to find sparse constraints, and it offers flexibility in choosing patterns P_i in a way that allows achieving a desired balance between the approximation quality and the complexity of (P-RLX). By choosing certain specific “shapes” of P_i , we can also prescribe sparse constraints of a particular structure. This is also possible in a dual setting. For example and as explained in detail later in Section 3.4.1, if we choose P_i of the form $\{2\alpha, \alpha + \beta, 2\beta\}$, we end up with a SDSOS relaxation. If we choose the P_i to be a simplicial circuit, we end up with a dual version of the SONC relaxation. There are many different approaches that can also be combined to formulate (P-RLX).

We believe that our template how to derive convexifications of (POP) via the formal problems (CVX-POP) and (P-RLX) has a definite methodological advantage. While other sparse approaches are discussed inseparably of modeling methods, we clearly separate modeling and sparsity. In the course of developing a particular approach to solve (POP) we can change a modeling strategy, but keep the sparsity-imposing strategy unchanged or vice versa. In other words, the ways of imposing sparsity need not be presented together with the modeling techniques. Both tasks can be coordinated, but there is a certain amount of independence in carrying out these two tasks. As

one of the highlights, we stress that the key ideas of duality theory for sparse relaxations are independent of the concrete modeling approach. We are not aware of any other source where the duality for (P-RLX) is presented in a model-independent way.

The procedure of developing convexifications without imposing sparsity can be described via a diagram,

$$X \xrightarrow{\text{monomial map}} \{(x^\alpha)_{\alpha \in A} : x \in X\} \xrightarrow{\text{convexifying}} \mathcal{M}_A(X) \xrightarrow{\text{lifting}} \mathcal{M}_B(X).$$

The lifting part $\mathcal{M}_A(X) \rightarrow \mathcal{M}_B(X)$ is needed, when we want to pick a larger exponent set $B \supseteq A$ that we perceive to be more convenient to work with than the original set A . For example, if $n = 1$, and we have $A = \{0, 2, 3, 6\}$, which means that we optimize a polynomial with the exponents x^0, x^2, x^3, x^6 , we might not have a specific strategy that would help optimizing such a polynomial over X . Instead, we might consider $\mathcal{M}_B(X)$ for $B = \{0, 1, 2, 3, 4, 5, 6\}$, which would help us optimize all polynomials of degree at most 6 over X . By providing a model for $\mathcal{M}_B(X)$ and projecting $\mathcal{M}_B(X)$ onto $\mathcal{M}_A(X)$ we establish a strategy for optimizing polynomials over X with the exponent vectors in A . Clearly, for solving the problem practically, we need to proceed with modeling and applying a particular convex optimization solver:

$$\mathcal{M}_B(X) \xrightarrow{\text{modeling}} \text{Model of } \mathcal{M}_B(X) \xrightarrow{\text{solving}} \dots$$

When establishing a model, lifting can be used once again, since sometimes $\mathcal{M}_B(X)$ is conveniently described as a projection of a higher-dimensional set.

The procedure of developing a sparse convexification is a variation of the above procedure. In the sparse version, the set B is the union of P_1, \dots, P_N , and rather than trying to model $\mathcal{M}_B(X)$ we model its N projections, so that the respective diagram is as follows:

$$\mathcal{M}_B(X) \xrightarrow{\text{projecting}} \mathcal{M}_{P_i}(X) \xrightarrow{\text{modeling}} \text{Model of } \mathcal{M}_{P_i}(X)$$

applied for each $i \in [N]$. When models are established, we can move over to solving:

$$\text{Model of } \mathcal{M}_{P_1}(X), \dots, \text{Model of } \mathcal{M}_{P_N}(X) \xrightarrow{\text{solving}} \dots$$

Figure 1 illustrates the procedure for the approximation of a cubic polynomial in one variable x and exponents $A = \{1, 2, 3\}$ on $X = [0, 1]$. Note that here we used $B = A$. In general, pattern relaxations can not be expected to be as tight as in this simple case.

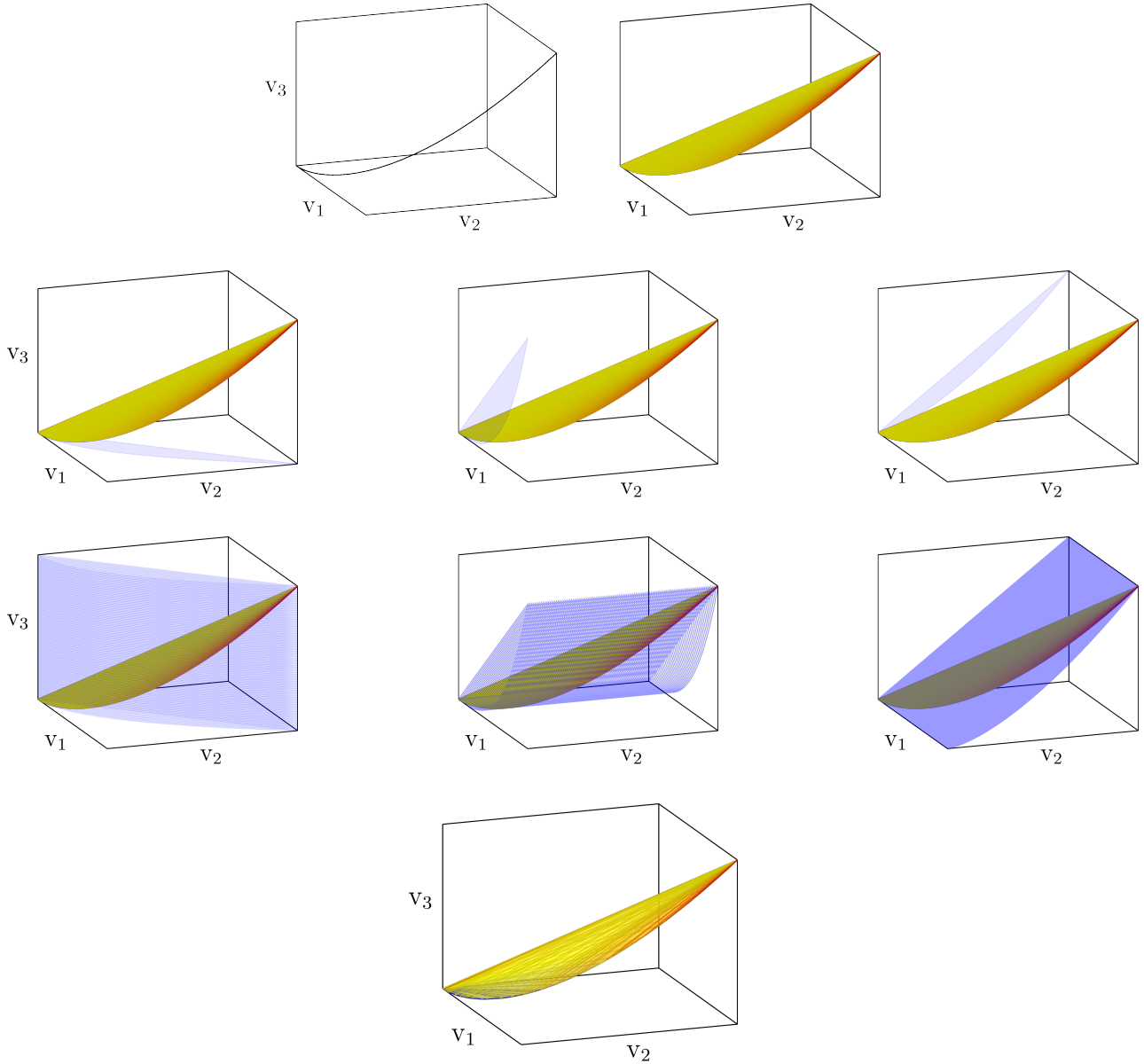
The above strategy is an ‘‘assembly line’’ that brings us from (POP) to a lower bound on (POP). It consists of the following steps and decisions:

1. What $B \supseteq A$ and which sets P_1, \dots, P_N covering B do we use?
2. How do we model $\mathcal{M}_{P_i}(X)$? In which optimization paradigm?
3. What algorithm is appropriate for the relaxation we have established?

The focus of our exposition is on the first question. We shall structure our review according to linear programming, second-order cone programming, and semidefinite programming where appropriate. The question of the overarching optimization paradigm and algorithmic choices will only be shortly commented upon in the following section.

1.4 Modeling choices and optimization paradigms

As already mentioned in Section 1.1.3, there are two schools of thought on computationally dealing with $\mathcal{M}_A(X)$, to which we shall refer shortly as SDP and NLP communities, respectively. SDP communities introduce a sequence of semidefinitely representable sets $(S_d)_{d \geq \deg f}$ that approximate $\mathcal{M}_A(X)$ and converge to $\mathcal{M}_A(X)$ as $d \rightarrow \infty$. Fixing a specific d , (CVX-POP) can be relaxed by replacing $\mathcal{M}_A(X)$ with S_d , and semidefinite programming algorithms can be used to solve the resulting relaxation. However, as d grows, the complexity of the underlying semidefinite relaxation may grow immensely. In practice, only the lowest level of the hierarchy is usually used in computations. The NLP community advocates the use of linear programming models. Linear programming models may be less suitable for finding tight approximations of the sets $\mathcal{M}_A(X)$, because $\mathcal{M}_A(X)$ may not be polyhedral and linear constraints can only describe polyhedra exactly. However, linear models work extremely well in branch-and-bound frameworks, because the dual simplex method for linear programming can be warm-started. There are additional approaches targeted towards other optimization paradigms. Usually these are other forms of cone programming, such as second-order cone or copositive programming. We do not intend to



■ **Figure 1** Elementary steps towards a moment relaxation of a one-dimensional cubic polynomial. **First row, left:** the curve $(x^\alpha)_{\alpha \in A}$ from (2) for $A = \{1, 2, 3\}$ and $X = [0, 1]$, indicating the feasible values for the lifted, monomial variables v . **Right:** the moment body $\mathcal{M}_A(X) = \text{conv}\{(x^\alpha)_{\alpha \in A} : x \in X\}$, see (3). **Second row, left to right:** projections $\mathcal{M}_{P_i}(X)$ of $\mathcal{M}_A(X)$ in blue for $P_i \in \mathcal{F} := \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$. **Third row, left to right:** liftings of $\mathcal{M}_{P_i}(X)$ into \mathbb{R}^3 for all $P_i \in \mathcal{F}$. **Last row:** the (intersected) feasible region $(v_\alpha)_{\alpha \in P_i} \in \mathcal{M}_{P_i}(X)$ for $i \in [3]$ of problem (P-RLX).

argue in favor of either of these schools. Also, it is beyond the scope of this review to look into the algorithmic details.

Our intention is to show the similarities of the convexification theory independent of the choice of a particular modeling approach and to integrate the available modeling techniques into the unifying convexification theory. In this sense we want to highlight one important similarity here. In all approaches, ultimately convex optimization problems have to be solved – either as subproblems in a divide-and-conquer approach, or as a reformulation of the original problem. For these convex optimization problems descent based algorithms can be applied, because a local optimum is also a global optimum. Often an interior point method is used, sometimes also active set based approaches as the simplex method. Now it is crucial from a practical point of view to understand the impact of different relaxations on the computational runtime. In general, this is very difficult to assess because of the unknown iteration numbers until convergence. However, the computational cost per iteration as an important contributor to the overall runtime is somewhat easier to estimate and a good indicator. In all Quasi-Newton

methods – primal or primal-dual interior point, active set based, or parametric programming – the computational cost per iteration is usually dominated by the costs to solve systems of linear equations.

The main motivation for sparsification as discussed in Section 1.3 is that sparsity can be exploited in the numerical linear algebra to solve systems of linear equations. E.g., the MOSEKhandbook [50] emphasizes that “having many small matrix variables is more efficient than one big matrix variable”. Some details on the specific case of interior point methods for large-scale cone programming are provided in [4]. The main reason for this is that it is computationally cheaper to solve several smaller systems of linear equations than fewer and larger ones. You would always prefer to invert 1000 matrices of size 10 by 10 in comparison to inverting one matrix of size 10000 by 10000. This is due to the cubic runtime of the Cholesky method to obtain a LU decomposition of a symmetric matrix. Numerical results for the solution of SDPs of varying size indicate the approximate cubic runtime increase in the SDP size [4, Table1.5]. Considering matrices of size n by n , a block decomposition into N matrices of size m would thus be asymptotically highly preferable if $Nm^3 \ll n^3$. Of course there are many more details. Linear matrix inequalities may contain several matrix variables, and matrix variables may feature in several inequalities, and all of them may have different sizes. Sparsity may not only be exploited via block decompositions, but in a variety of ways by exploiting specifics of the model or of the optimization paradigm [4]. In some nonlinear optimization solvers, BFGS updates are directly applied to an approximation of the inverse of the Hessian, reducing the effort from cubic to quadratic for a matrix vector multiplication.

Still, as a rule of thumb the rough $Nm^3 \ll n^3$ improvement motivates why it is so interesting to look at the intersection of many patterns $\mathcal{M}_{P_i}(X)$ of reduced size compared to $\mathcal{M}_A(X)$. One illustrative example was shown in Figure 1, although here no significant savings can be expected for the small dimensions $N = n = 3, m = 2$. As a main take-away we summarize that in all approaches to solving (POP), the runtime of algorithms can be reduced drastically by creating and exploiting sparsity. In practice, this often makes the difference between intractable and computationally feasible and is a driving force in polynomial optimization research.

1.5 Duality

There are two dual points of view on polynomial optimization: the convexification point of view we have just discussed and the non-negativity certification point of view. Non-negativity certification is a far-reaching generalization of the ideas behind Farkas’ lemma and duality in linear programming to the case of polynomial optimization. When one presents the theory of polynomial optimization, one either starts with the convexification and then passes through non-negativity certification via dualization or one does it the other way around, starting with the questions of non-negativity certification and then passing to convexification via dualization. We believe that it is instructive to have a general presentation of the duality theory for sparse approaches, purified from the modeling details. Such a presentation allows understanding the core ideas and to gain insight regarding how different approaches are related to each other. For many specific choices of patterns, duality theorems were formulated independently of each other. The general duality underpinning all of these special cases is the duality of (P-RLX) to the bound-certification problem

$$\sup\{\lambda \in \mathbb{R} : f - \lambda = g_1 + \dots + g_N, g_i \in \mathcal{P}(X)_{P_i} \ i \in [N]\}. \quad (4)$$

Here, the non-negativity of $f - \lambda$ on X is to be certified as a sum of sparse polynomials with supports contained in P_1, \dots, P_N which are non-negative on X . Analogously to how we discussed modeling of (P-RLX), there is an aspect of modeling $\mathcal{P}(X)_{P_i}$ coming up in relation to the bound-certification problem. Usually, a model of $\mathcal{M}(X)_{P_i}$ can be dualized to obtain a model of $\mathcal{P}(X)_{P_i}$, applying the duality of the respective optimization paradigm (e.g., linear, second-order cone or semidefinite-programming duality). Special cases, and a detailed discussion of duality will follow in Section 3.

1.6 Contributions

We emphasize that none of the propositions in this review is genuinely novel or surprising, although we formulated them in a self-consistent manner using basis tools from convex analysis and linear algebra. In particular the proofs for Theorems 4 and 11 are more compact than in other references we are aware of. The main contribution lies in generalization and abstraction of existing ideas. The unifying framework allows the derivation of novel approaches to convexification and/or sparsification, but also to derive theoretical results applicable to all existing and future methods. We illustrate the potential of convexification and sparsification with numerical experiments.

1.7 Structure of the paper

In Section 2 we provide an overview of two basic convexification principles, based on linear and semidefinite programming, respectively, and provide concrete suggestions on how to flexibly sparsify the convexification approaches. The second major part of this paper addresses in Section 3 the conic viewpoint to pattern relaxations, duality theory for pattern relaxations, and the theory of lifted conic formulations in the context of polynomial optimization. This part provides a general systematic treatment of duality for sparse approaches to polynomial optimization. It is independent from Section 2. In the Appendix we provide a list of notation, numerical experiments for specific patterns, and a short discussion how sparsity may be addressed in current solvers.

2 Modeling principles for convexification

In this section we shall review different techniques for convexification and sparsification of $\{(x^\alpha)_{\alpha \in A} : x \in X\}$, guided by different principles. We start with some definitions and a look at semidefinite and linear programming, before we continue with sparsification.

Let \mathbb{S}^m be the space of symmetric matrices of size m over reals and \mathbb{S}_+^m the convex cone of positive semidefinite matrices in \mathbb{S}^m . For a matrix $M \in \mathbb{S}^m$, we denote the condition $M \in \mathbb{S}_+^m$ as $M \succeq 0$.

If $M : \mathbb{R}^N \rightarrow \mathbb{S}^m$ is an affine map, then the condition $M(v) \succeq 0$ is called a linear matrix inequality (LMI) of size m . In other words, $M(v)$ is a symmetric matrix with the entries being affine functions in v and the condition $M(v) \succeq 0$ requires positive semidefiniteness of $M(v)$. Semidefinite optimization is optimization of a linear objective function subject to finitely many LMIs. If all the LMIs have size 2, then we get a special case of semidefinite optimization, called second-order cone optimization. If all LMIs have size 1, we obtain linear programming as an even more special case of semidefinite optimization. Semidefinite, second-order cone, and linear optimization have a well-developed algorithmic theory, with the underlying algorithms implemented in a huge number of solvers and used in myriads of applications.

► **Definition 1.** *The set defined by finitely many LMIs is called a spectrahedron. A set which is the image of a spectrahedron under a linear transformation is called a projected spectrahedron or a semidefinitely representable set.*

2.1 Semidefinite modeling principles

We first look at semidefinite modeling principles.

2.1.1 Semidefinite convexifications from inference rules

Our objective is to lay out ways of deriving convex inequalities valid on $\mathcal{M}_A(X)$ for the feasible set X defined in (1). Observe that, if $p(x) \geq 0$ is a polynomial inequality valid for $x \in X$, then $L_v(p) \geq 0$ is a linear inequality valid for $v \in \mathcal{M}_A(X)$. Furthermore, if $M(x)$ is a matrix whose entries are polynomials in $\mathbb{R}[x]_A$ and the condition $M(x) \succeq 0$ is true for $x \in X$, then the LMI $L_v(M(x)) \succeq 0$ is true for $v \in \mathcal{M}_A(X)$, where $L_v(M(x))$ is the entry-wise linearization of the matrix $M(x)$.

Let $M_B(x) = x^B(x^B)^\top$. For every $x \in \mathbb{R}^n$, we have easy inference rules:

$$p_1(x) \geq 0, \dots, p_t(x) \geq 0 \quad \Rightarrow \quad p_1(x) \dots p_t(x) M_B(x) \succeq 0, \quad (5)$$

$$p_1(x) \geq 0, \dots, p_t(x) \geq 0 \quad \Rightarrow \quad p_1(x) \dots p_t(x) \geq 0, \quad (6)$$

$$p(x) \geq 0 \quad \Rightarrow \quad p(x) M_B(x) \succeq 0, \quad (7)$$

$$1 \geq 0 \quad \Rightarrow \quad M_B(x) \succeq 0, \quad (8)$$

$$p(x) = 0 \quad \Rightarrow \quad p(x) q(x) = 0 \quad (9)$$

where $p, p_1, \dots, p_t, q \in \mathbb{R}[x]$ and B is a finite subset of \mathbb{N}^n . The inequalities $g_i(x) \geq 0$ and the equations $h_j(x) = 0$ that define X are our fundamental knowledge concerning feasibility. Application of the inference rules gives us further, redundant knowledge, which turns out to be useful for convexification purposes. Rules (5–8) deal with inequalities and differ in their degree of generality, with the rule (5) subsuming the rules (6–8). (6) is a special case of (5) with $B = \{0\}$, (7) is a special case of (5) with $t = 1$ and (8) is a special case of (7) with $p = 1$.

Inserting the polynomial inequalities and equations defining X into the premises of (5–8) and linearizing the conclusions of these rules using L_v , we obtain linear and semidefinite constraints that are valid for the moment

body of X . In this way different outer relaxations of the moment body $\mathcal{M}_A(X)$ can be derived as projected spectrahedra. If one only uses the rule (6), one obtains an outer approximation by a polyhedron.

The more general the rule is, the harder the implied constraints for $\mathcal{M}_A(X)$ may become. For example, if the product $p_1(x) \dots p_t(x)$ in rules (5) and (6) has many factors, then the respective linearization will involve many monomial variables. If an exponent set B involved in rules (5) and (7) is large, then the respective linearization will be an LMI of a large size. That is, more general rules allow a more exact approximation of $\mathcal{M}_A(X)$, but come at a higher computational price as discussed in Section 1.4.

► **Remark 2.** Let us estimate the number of monomial variables involved in the linearization derived from (5–9). Since in our context linearization is simply substitution of monomials with monomial variables, we need to track the relation between the monomials involved in the premises and the conclusions of our inference rules.

- If $p_i(x) \in \mathbb{R}[x]_{A_i}$, then the entries of the matrix $p_1(x) \dots p_t(x) M_B(x)$ in the conclusion of (5) belong to $\mathbb{R}[x]_{A_1 + \dots + A_t + B + B}$.
- For (6), one has $p_1(x) \dots p_t(x) \in \mathbb{R}[x]_{A_1 + \dots + A_t}$.
- For (7), if $p(x) \in \mathbb{R}[x]_C$, the entries of $p(x) M_B(x)$ belong to $\mathbb{R}[x]_{C + B + B}$.
- For (8), the entries of $M_B(x)$ belong to $\mathbb{R}[x]_{B + B}$.
- For (9), if $p(x) \in \mathbb{R}[x]_C$ and $q(x) \in \mathbb{R}[x]_D$, then $p(x)q(x) \in \mathbb{R}[x]_{C + D}$.

Deliberately choosing exponent sets A_i, B, C, D of a manageable size and appropriate shape, we can control the computational complexity associated with the processing of the respective linearizations. That is, we can make the constraints sparse and we can produce LMIs of a desirable (and not too large) size. This is an overarching idea of most of the existing convexifications that rely on rules (5–9).

2.1.2 Positivstellensätze and their duals

Real algebra and real algebraic geometry allow us to understand in which cases the convexifications form the above inference rules approximate $\mathcal{M}_A(X)$ arbitrarily well or even describe $\mathcal{M}_A(X)$ exactly. From the practical perspective, however, one definitely needs to keep track of the complexity of the convex constraints resulting from those rules. The inferences (7) and (8) are the favorite choices for linearization in the semidefinite optimization and real-algebra communities. These choices are backed by Putinar’s Positivstellensatz [58], which uses so-called sum of squares certificates of positivity. A polynomial in $\mathbb{R}[x]$ is called SOS, if it admits a representation $f_1^2 + \dots + f_t^2$ with $f_1, \dots, f_t \in \mathbb{R}[x]$. We denote by Σ the set of all SOS polynomials in $\mathbb{R}[x]$ and call it the infinite-dimensional SOS cone.

► **Theorem 3 (Putinar).** *Let $X = \{x \in \mathbb{R}^n : g_1(x) \geq 0, \dots, g_s(x) \geq 0\}$ be a compact semialgebraic set contained in the ball $\{x : x_1^2 + \dots + x_n^2 \leq \rho\}$ of some radius $\rho \in \mathbb{R}_{>0}$. Let $g_0(x) := 1$ and $g_{s+1}(x) := \rho - (x_1^2 + \dots + x_n^2)$. Then every polynomial f which is strictly positive on X belongs to $f = g_0 \Sigma + \dots + g_{s+1} \Sigma$.*

In the following theorem, we use the infinite symmetric positive semidefinite matrix $M_{\mathbb{N}^n} = x^{\mathbb{N}^n} (x^{\mathbb{N}^n})^\top$. For a symmetric matrix $M = (m_{i,j})_{i,j \in I} \in \mathbb{R}^{I \times I}$ with an infinite index set I , we define positive semidefiniteness condition $M \succeq 0$ as the validity of $(m_{i,j})_{i,j \in B}$ for every finite $B \subseteq I$. We use the following notation: if $y \in \mathbb{R}^n$ and $B \subseteq \mathbb{N}^n$, then for $A \subseteq B$ the vector y_A is the projection of y on the coordinates indexed by A : $y_A = (y_\alpha)_{\alpha \in A} \in \mathbb{R}^A$. The set Y_A denotes the projection of a set Y on the A coordinates, i.e., $Y_A = \{y_A : y \in Y\}$.

► **Theorem 4 (Dual Putinar).** *In the notation of Theorem 3, for every finite set $A \subseteq \mathbb{N}^n \setminus \{0\}$, one has*

$$\mathcal{M}_A(X) = \{v \in \mathbb{R}^{\mathbb{N}^n} : v_0 = 1, L_v(g_i M_{\mathbb{N}^n}) \succeq 0 \text{ for all } i = 0, \dots, s + 1\}_A. \quad (10)$$

Proof. The set $\mathcal{M}_A(X)$ is the closed convex hull of $\{(p^\alpha)_{\alpha \in A} : p \in X\}$. Hence, for showing the part “ \subseteq ” of the set equality (10), it is enough to check that each $w = (p^\alpha)_{\alpha \in A}$ with $p \in X$ belongs to the right-hand side of (10). For $v = (p^\alpha)_{\alpha \in \mathbb{N}^n}$ we have $v_A = w$, $v_0 = 1$ and $L_v(g_i M_{\mathbb{N}^n}) = g_i(p) M_{\mathbb{N}^n}(p)$, where $g_i(p) \geq 0$ and $M_{\mathbb{N}^n}(p) = vv^\top$ is an infinite rank-one positive semidefinite matrix. This shows that w belongs to the right-hand side of (10). To conclude the proof of (10), we consider an arbitrary $w \in \mathbb{R}^A \setminus \mathcal{M}_A(X)$ and show that such w does not belong to the right-hand side of (10). By separation theorem, there exists $c \in \mathbb{R}^{A \cup \{0\}}$ such that $\sum_{\alpha \in A} w_\alpha c_\alpha + c_0 < 0$ and $\sum_{\alpha \in A} u_\alpha c_\alpha + c_0 \geq 0$ for every $u \in \mathcal{M}_A(X)$. Taking $u_\alpha = p^\alpha$ with $p \in X$, we see that the polynomial $\sum_{\alpha \in A} x^\alpha c_\alpha + c_0$ is non-negative on X . Fix $\varepsilon > 0$ small enough to satisfy $\sum_{\alpha \in A} w_\alpha c_\alpha + c_0 + \varepsilon < 0$. Since $f = \sum_{\alpha \in A} c_\alpha x^\alpha + c_0 + \varepsilon$ is strictly positive on X , by Theorem 3 we conclude that $f = g_0 \sigma_0 + \dots + g_{s+1} \sigma_{s+1}$,

where $\sigma_0, \dots, \sigma_{s+1} \in \Sigma$. Consider any vector $v \in \mathbb{R}^{\mathbb{N}^n}$ satisfying $v_A = w$. We have

$$0 > \sum_{\alpha \in A} c_\alpha w_\alpha + c_0 + \varepsilon = L_v(f) = \sum_{i=0}^s L_v(g_i \sigma_i).$$

Hence $L_v(g_i \sigma_i) < 0$ for some $i \in \{0, \dots, s\}$. Since σ_i is a sum of squares, for some of these squares, which we denote as q^2 , one has $L_v(g_i q^2) < 0$. Let q be given as $q = s^\top x^B$ for some finite $B \subseteq \mathbb{N}^n$ and some $s \in \mathbb{R}^B$. Then $g_i q^2 = s^\top g_i M_B s$ and by this, $L_v(g_i q^2) = s^\top L_v(g_i M_B) s < 0$. The latter shows that $g_i M_B$ is not positive semidefinite, which implies that the condition $g_i M_{\mathbb{N}^n} \succeq 0$ is violated. Consequently, w is not in the right-hand side of (10). \blacktriangleleft

► **Remark 5.** A natural question arises about the meaning of the set on the right-hand side of (10), of which we take the projection onto \mathbb{R}^A . Or in other words, what would be the case $A = \mathbb{N}^n$ of Theorem 4? On the left-hand side as $\mathcal{M}_{\mathbb{N}^n}(X)$ one can expect some generalization of the closed convex hull operation for the set $\{x^{\mathbb{N}^n} : x \in X\}$. It turns out that the correct generalization is the system of “complete moment vectors” with respect to a probability measure concentrated on X . Such a measure-theoretic generalization of Theorem 4 is nice to have, but it comes with the price of being more technical, as measure-theoretic technicalities have to be addressed to properly define $\mathcal{M}_{\mathbb{N}^n}(X)$. We would like to stress that Theorem 4, which does not involve any measure-theoretic machinery, is already enough to establish the convergence of the hierarchy of moment relaxations as outlined in the following section.

2.1.3 Semidefinite hierarchies of moment relaxations

We can establish sequences of approximations of polynomial optimization problems. The approximations are based on either linear or semidefinite convex problems and have convergence guarantees from the theorems of the previous section. We discuss these hierarchies and approximations with sparsity in mind.

Consider the setting of Theorem 4, assuming that we know an A_i with $g_i \in \mathbb{R}[x]_{A_i}$ for $i = 0, \dots, s+1$. Each constrained $L_v(g_i M_{\mathbb{N}^n}) \succeq 0$ can be “cropped” to a constraint $L_v(g_i M_{B_i}) \succeq 0$, by fixing a finite set $B_i \subseteq \mathbb{N}^n$. This is an LMI of a finite size $|B_i|$. In order to take into consideration of all the monomial variables v_α with $\alpha \in A$ in constraints it suffices to have $B_0 + B_0 \supseteq A$, since the constraint $L_v(g_0 M_{B_0}) = L_v(M_{B_0}) \succeq 0$ involves the variables v_β with $\beta \in B_0 + B_0$. Hence, each choice of B_0, \dots, B_{s+1} produces an outer approximation $\text{MR}(B_0, \dots, B_{s+1})_A$ of $\mathcal{M}_B(X)$, where with $P := \bigcup_{i=0}^{s+1} (A_i + B_i + B_i)$ one has

$$\text{MR}(B_0, \dots, B_{s+1}) := \{v \in \mathbb{R}^P : L_v(g_i M_{B_i}) \succeq 0 \text{ for all } i = 0, \dots, s+1\}. \quad (11)$$

The particular choice of the B_i has a strong impact on the resulting relaxation with a trade-off between computational costs and tightness of bounds. One can also interpret the choice of B_i as a step in a theoretical proof of approximation properties. In practice, the choice of B_i is also closely connected to the question how the relaxed optimization problem can be solved algorithmically. We shall see examples resulting in linear or semidefinite optimization problems later. While algorithms are not the main focus of our review, we want to emphasize that all particular approaches in the literature can be seen as special cases of this generic proof-centered framework.

With the definition (11), $\text{MR}(B_0, \dots, B_{s+1})_A$ is a projected spectrahedron in \mathbb{R}^A that approximates the moment body $\mathcal{M}_A(X)$ and is given as a projection of a spectrahedron in \mathbb{R}^P . When all of the B_i converge to \mathbb{N}^n , then $\text{MR}(B_0, \dots, B_{s+1})_A$ converges to $\mathcal{M}_A(X)$ in the so-called Hausdorff metric. We recall that for compact sets K, L , the Hausdorff metric is defined by $\delta(K, L) = \max\{\max_{q \in L} \min_{p \in K} \|p - q\|, \max_{p \in K} \min_{q \in L} \|p - q\|\}$. When $K \subseteq L$, this is the minimal $\rho \geq 0$ such that every point of K is at distance at most ρ to some point of L .

► **Corollary 6.** *In the notation of Theorem 4, if $(B_{i,t})_{t \in \mathbb{N}}$ with $i = 0, \dots, s+1$ are sequences of finite subsets of \mathbb{N}^n that are increasing in t with respect to inclusion and satisfying $\bigcup_{t \in \mathbb{N}} B_{i,t} = \mathbb{N}^n$ for all $i \in \{0, 1, \dots, s+1\}$, then*

$$\lim_{t \rightarrow \infty} \text{MR}(B_{0,t}, \dots, B_{s+1,t})_A = \mathcal{M}_A(X)$$

holds in the sense of the Hausdorff metric.

Proof. Using $\bigcup_{t \in \mathbb{N}} B_{i,t} = \mathbb{N}^n$ and Theorem 4, we obtain

$$\bigcap_{t \in \mathbb{N}} \text{MR}(B_{0,t}, \dots, B_{s+1,t})_A = \mathcal{M}_A(X). \quad (12)$$

Since $B_{i,t}$'s are increasing in t with respect to inclusion, the set $\text{MR}(B_{0,t}, \dots, B_{s+1,t})_A$ is decreasing in t with respect to inclusion. Thus, applying Lemma 1.8.2 from [63] to (12), we get the assertion. \blacktriangleleft

► **Remark 7.** Lasserre introduced the canonical choice $B_{i,t} = \mathbb{N}_{2(d-d_i+t)}^n$ in the case, where $A = \mathbb{N}_{2d}^n$, $B_i = \mathbb{N}_{2d_i}^n$ and $d \geq d_i$. For this choice $A_i + B_{i,t} = \mathbb{N}_{2(d+t)}^n$ so that the spectrahedron $\text{MR}(B_{0,t}, \dots, B_{s+1,t})$ is a subset of $\mathbb{R}^{\mathbb{N}_{d+t}^n}$.

► **Remark 8.** Theorem 6 alludes to an iterative method for solving the problems $\inf_{x \in X} f(x)$, where $f \in \mathbb{R}[x]_A$. Such a method would iteratively increment t and solve a semidefinite optimization problem on $R_t := \text{MR}(B_{0,t}, \dots, B_{s+1,t})$ in each iteration in order to provide an improved lower bound on $\inf_{x \in X} f(x)$, which would converge to the true value as $t \rightarrow \infty$. However, in practice, such a method would be extremely impractical. Traditionally, one uses interior-point methods for solving SDPs which are difficult to warm-start. In addition, the size of the LMIs describing R_t get larger (and quite dramatically so for large n) as t increases, leading to increased run times as discussed in Section 1.4. Instead of practically using Theorem 6, one could treat it as a framework result showing that, in principle, an arbitrarily precise approximation of $\inf_{x \in X} f(x)$ is possible, albeit usually at a very high cost. The fact that the price to be paid for accuracy is very high is not very surprising, because solving polynomial optimization problems to global optimality is known to be computationally hard.

The practical conclusion we make here is that we should choose sets B_0, \dots, B_{s+1} that define the approximation $\text{MR}(B_0, \dots, B_{s+1})_A$ carefully. They should not be too large and the structure of the particular problem $\inf_{x \in X} f(x)$ should be exploited, such that the resulting SDPs can be solved in reasonable time.

► **Remark 9.** A natural modification of the relaxation $R(B_0, \dots, B_{s+1})_A$ is obtained by replacing the semidefinite constraint $L_v(g_i M_{B_i}) \succeq 0$ with a family of constraints $L_v(g_i M_B)$ where B belongs to a given family \mathcal{B}_i of finite subsets of \mathbb{N}^n . In this way, one large B_i can be replaced by smaller B 's taken from \mathcal{B}_i . This new relaxation might provide a looser approximation, but it may become more manageable from a computational perspective. The relaxation is $\text{MR}(\mathcal{B}_0, \dots, \mathcal{B}_{s+1})_A$, where

$$\text{MR}(\mathcal{B}_0, \dots, \mathcal{B}_{s+1}) := \{v \in \mathbb{R}^P : L_v(g_i M_B) \succeq 0 \text{ for all } i = 0, \dots, s+1, \text{ and } B \in \mathcal{B}_i\}.$$

Just as with the choice of the B_i , choosing a particular family \mathcal{B}_i is typically balancing theoretical guarantees for the bounds and algorithmic efficiency.

2.2 Linear programming models

Now we look at different convexification approaches that result in linear programming models. Note that we often assume specific structures $X = K$ of the feasible set such as K being a polytope or $K = \text{Box}(l, u)$.

2.2.1 Bound-factor relaxations

Another useful positivstellensatz goes back to Handelman [29].

► **Theorem 10 (Handelman).** *Let K be a non-empty polytope given as $K = \{x \in \mathbb{R}^n : g_1(x) \geq 0, \dots, g_s(x) \geq 0\}$, where $g = (g_1, \dots, g_s)$ are polynomials of degree one. Then every polynomial $f \in \mathbb{R}[x]$ strictly positive on K admits a representation*

$$f = \sum_{\beta \in B} c_\beta g^\beta,$$

where $B \subseteq \mathbb{N}^s$ is a finite set and $c_\beta \geq 0$ for every $\beta \in B$.

► **Theorem 11 (Dual Handelman).** *In the notation of Theorem 10, for every finite set $A \subseteq \mathbb{N}^s \setminus \{0\}$, one has*

$$\mathcal{M}_A(K) = \{v \in \mathbb{R}^{\mathbb{N}^s} : v_0 = 1, L_v(g^\beta) \geq 0 \text{ for all } \beta \in \mathbb{N}^s\}_A. \quad (13)$$

Proof. The proof is analogous to the proof of the dual Putinar theorem 4, but somewhat simpler, because it involves linear constraints rather than the semidefinite ones. To show the inclusion " \subseteq ", we pick p , w and v in the same way as in the proof of dual Putinar (note that K in the dual Handelman corresponds to X in the dual Putinar). Then $L_v(g^\beta) = g(v)^\beta \geq 0$, since $g_i(v) \geq 0$ for each $i \in [s]$ in view of the fact that v belongs to P . This shows that w is in the right-hand side of (13).

To show the inclusion “ \supseteq ”, we assume that w is in $\mathbb{R}^A \setminus \mathcal{M}_A(K)$. Then, as in the proof of the dual Putinar, we fix a separating hyperplane determined by the vector c , introduce an $\varepsilon > 0$ such that the polynomial $f(x)$ (depending on c and ε) is strictly positive on K . By Handelman, $f = \sum_{\beta \in B} d_\beta g^\beta$, where B is a finite subset of \mathbb{N}^s and d_β are non-negative coefficients.

For every $v \in \mathbb{N}^n$ with $v_B = w$, we have

$$\begin{aligned} 0 &> \sum_{\alpha \in A} c_\alpha w_\alpha + \varepsilon \\ &= L_v(f) = \sum_{\beta \in B} d_\beta L_v(g^\beta). \end{aligned}$$

This implies that for some $\beta \in B$ we have $L_v(g^\beta) < 0$. Consequently, v is not in the right-hand side set of (13). ◀

Here, we work under the assumptions of Theorem 10. Consider $0 \in A_i \subseteq \mathbb{N}_1^n$ such that $g_i \in \mathbb{R}[x]_{A_i}$. We can also fix $A_i = \mathbb{N}_1^n$, which is the case needed when the coefficients of g_i are generic, but in more specific situations, we might prefer to take a smaller A_i . We introduce the polyhedral relaxation $\text{BF}(B)$ of $\mathcal{M}_A(K)$ as

$$\text{BF}(B) := \{v \in \mathbb{R}^P : L_v(g^\beta) \geq 0 \text{ for all } \beta \in B\} \tag{14}$$

depends on the choice of a finite set $B \subseteq \mathbb{N}^s$ and

$$P := \bigcup_{\beta \in B} (\underbrace{A_1 + \dots + A_1}_{\beta_1} + \dots + \underbrace{A_s + \dots + A_s}_{\beta_s}).$$

We call $\text{BF}(B)$ the bound-factor polyhedron for B . Analogously to how we analyzed established semidefinite relaxations in Corollary 6, one can show that $\text{BF}(B)$ can approximate $\mathcal{M}_A(K)$ arbitrarily well. In particular, $\lim_{d \rightarrow \infty} \text{BF}(\mathbb{N}_d^s)_A = \mathcal{M}_A(K)$ with the limit in the sense of the Hausdorff metric.

► **Remark 12.** In contrast to linear programming relaxations, approximation by semidefinite relaxations is achievable for a much broader class of moment bodies $\mathcal{M}_A(X)$, where X is not necessarily a polytope. In some cases (for example, for univariate problems) one can even guarantee an exact description of $\mathcal{M}_A(X)$. In sharp contrast to this, an exact polyhedral description is impossible with linear programming relaxations if $\mathcal{M}_A(X)$ is not a polyhedron. On the other hand, linear programming relaxations are better suited for applications in branch-and-bound frameworks, because the dual simplex method can start in phase 2 with the solution of the parent node.

2.2.2 Exact polyhedral models

Since global NLP heavily relies on linear programming, it uses patterns with polyhedral moment bodies. Below, we present some such patterns.

► **Proposition 13.** *Let $I \subseteq \{0, 1\}^n$ and $K = \text{Box}(l, u)$. Then the moment body $\mathcal{M}_I(K)$ is a polytope given by $\mathcal{M}_I(K) = \mathcal{M}_I(\text{vert}(K))$.*

Proof. Clearly, $\text{vert}(K) = \{l_1, u_1\} \times \dots \times \{l_n, u_n\}$. By rearranging the identity $x_i(u_i - l_i) = x_i(u_i - l_i)$, one has

$$x_i = \underbrace{\frac{u_i - x_i}{u_i - l_i}}_{a_{i,l_i}} l_i + \underbrace{\frac{x_i - l_i}{u_i - l_i}}_{a_{i,u_i}} u_i.$$

with $a_{i,l_i}, a_{i,u_i} \geq 0$, for every $x \in K$, and $a_{i,l_i} + a_{i,u_i} = 1$. Representing x_1, \dots, x_n as above and using the fact that $m(x_1, \dots, x_n) := x^I$ is an affine function in each of the x_i 's, we obtain

$$m(x) = \sum_{s_1 \in \{l_1, u_1\}, \dots, s_n \in \{l_n, u_n\}} a_{1,s_1} \dots a_{n,s_n} m(s_1, \dots, s_n),$$

where all 2^n products $a_{1,s_1} \dots a_{n,s_n}$ are non-negative, for $x \in K$, and

$$\sum_{s_1 \in \{l_1, u_1\}, \dots, s_n \in \{l_n, u_n\}} a_{1,s_1} \dots a_{n,s_n} = (a_{1,l_1} + a_{1,u_1}) \dots (a_{n,l_n} + a_{n,u_n}) = 1.$$

This shows $\mathcal{M}_I(K) = \mathcal{M}_I(\text{vert}(K))$, which immediately implies that $\mathcal{M}_I(K)$ is a polytope. ◀

► **Remark 14.** Since $\mathcal{M}_I(K) = \mathcal{M}_I(\text{vert}(K))$, the condition in Proposition 13 that $v \in \mathcal{M}_I(K)$ can be formulated using the linear constraints

$$\begin{aligned} v &= \sum_{p \in \text{vert}(K)} \lambda_p m(p), \\ 1 &= \sum_{p \in \text{vert}(K)} \lambda_p \end{aligned}$$

with the non-negative auxiliary variables $\lambda_p \geq 0$.

► **Remark 15.** The rectangle $K = [l_1, u_1] \times [l_2, u_2]$ is given by $g = (g_1, g_2, g_3, g_4) = (x_1 - l_1, u_1 - x_1, x_2 - l_2, u_2 - x_2)$ with $g_i \in \mathbb{R}[x]_{A_i}$ and $A_1 = A_2 = \{(0, 0), (1, 0)\}$ and $A_3 = A_4 = \{(0, 0), (0, 1)\}$. In this case, it is known that for

$$B = \{(1, 0), (0, 1)\}^2 = \{(1, 0, 1, 0), (1, 0, 0, 1), (0, 1, 1, 0), (0, 1, 0, 1)\},$$

the bound-factor polyhedron $\text{BF}(B)$ coincides with the moment body $\mathcal{M}_A(K)$ for $A = \{0, 1\}^2$. The four inequalities describing $\text{BF}(B)$ are $L_v(g_1 g_3) \geq 0$, $L_v(g_1 g_4) \geq 0$, $L_v(g_2 g_3) \geq 0$, and $L_v(g_2 g_4) \geq 0$. They are known as McCormick inequalities.

► **Example 16.** Consider $\mathcal{M}_{\{0,1\}^2}(K)$ for $K = [-1, 1]^2$. By Proposition 13, this moment body is a polytope with the vertices $x^{\{0,1\}^2}$, where $x \in \{-1, 1\}^2$, so that $v \in \mathcal{M}_{\{0,1\}^2}(K)$ can be formulated as the constraint

$$\begin{pmatrix} 1 \\ v_{1,0} \\ v_{0,1} \\ v_{1,1} \end{pmatrix} = \lambda_1 \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} + \lambda_3 \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} + \lambda_4 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

using auxiliary non-negative variables $\lambda_1, \dots, \lambda_4 \geq 0$. Using McCormick's inequalities we can also formulate $\mathcal{M}_{\{0,1\}^2}(K)$ in the original space by

$$\begin{aligned} L_v((1 - x_1)(1 - x_2)) &:= 1 - v_{1,0} - v_{0,1} + v_{1,1} \geq 0, \\ L_v((1 + x_1)(1 - x_2)) &:= 1 + v_{1,0} - v_{0,1} - v_{1,1} \geq 0, \\ L_v((1 - x_1)(1 + x_2)) &:= 1 - v_{1,0} + v_{0,1} - v_{1,1} \geq 0, \\ L_v((1 + x_1)(1 + x_2)) &:= 1 + v_{1,0} + v_{0,1} + v_{1,1} \geq 0. \end{aligned}$$

As a next step, we generalize Proposition 13 by passing from $I = \{0, 1\}^n$ to a more general set P :

► **Proposition 17.** *Let $\alpha \in \mathbb{N}^n$, $P \subseteq \{0, \alpha_1\} \times \dots \times \{0, \alpha_n\}$, and $K = \text{Box}(l, u)$. Then the moment body $\mathcal{M}_P(K)$ is a polytope.*

Proof. There is no loss of generality in assuming $\alpha_i > 0$ for every $i \in [n]$. We can describe P as $P = \Gamma I$, for some $I \subseteq \{0, 1\}^n$ and the diagonal matrix $\Gamma \in \mathbb{N}^{n \times n}$ with the diagonal entries $\alpha_1, \dots, \alpha_n$. By Proposition 20 from Section 2.3, one has $\mathcal{M}_P(K) = \mathcal{M}_{\Gamma I}(K) = \mathcal{M}_I(K^\Gamma)$, where $K^\Gamma = \text{Box}(l_\Gamma, u_\Gamma)$. By Proposition 13, $\mathcal{M}_I(K^\Gamma)$ is a polytope. ◀

► **Remark 18.** Remark 14 and the proof of Proposition 17 yield a way to provide an explicit formulation of $\mathcal{M}_P(K)$ with $P \subseteq \{0, \alpha_1\} \times \dots \times \{0, \alpha_n\}$.

Motivated by Proposition 17, we call a pattern $P \subseteq \{0, \alpha_1\} \times \dots \times \{0, \alpha_n\}$ multilinear.

We also generalize Proposition 17 as follows.

► **Proposition 19.** *Let $\Gamma = (\gamma(1), \dots, \gamma(k)) \in \mathbb{R}^{n \times k}$ be a matrix, for which the monomials $x^{\gamma(i)}$ with $i \in [k]$ are variable-independent. Then for any $P \subseteq \Gamma \{0, 1\}^k$, the moment body $\mathcal{M}_P(K)$ is a polytope.*

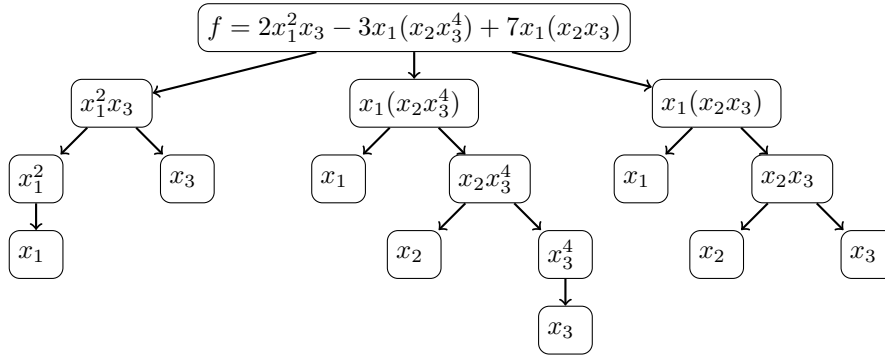
Proof. This is a direct consequence of Propositions 20 and 13. ◀

Patterns from expression trees

We describe how expression-tree convexifications from global NLP can be phrased in terms of patterns, when applied to (POP). Each algebraic expression is made up of a certain set of elementary operations, such as powers, linear combinations, or products of expressions. A decomposition of an algebraic expression into these operations

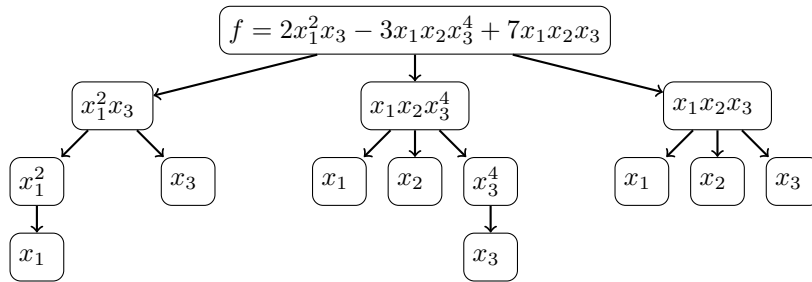
can be visualized using an algebraic expression tree. This is a rooted tree with nodes labeled by terms occurring in the expression. Each term is built up from its child terms using elementary operations. The underlying convexification is obtained by introducing a variable for each sub-expression occurring in the tree and providing convex constraints that link the variables of every node and its child nodes. A polynomial $f = \sum_{\alpha \in A} f_{\alpha} x^{\alpha}$ can thus be expressed as a linear combination of monomials, where each monomial x^{α} is expressed as a product of powers of the variables. If we do not want to use products with arbitrary number of factors as elementary operations, we can fix a bracketing in the product $x_1^{\alpha_1} \dots x_n^{\alpha_n}$, to specify the order in which multiplications are carried out.

So, if $f = 2x_1^2x_3 - 3x_1(x_2x_3^4) + 7x_1(x_2x_3)$, we have the corresponding tree



We assign monomial variables to all expressions apart from the root node f . The convexification for the root node f and the variables $v_{2,1,0}$, $v_{1,1,4}$, and $v_{1,1,1}$ assigned to its children is $f = 2v_{2,1,0} - 3v_{1,1,4} + 7v_{1,1,1}$. Every non-root node of a tree gets assigned a monomial variable. Taking a non-root node together with its children determines a pattern. For example, the node $x_1(x_2x_3^4)$ and its children form the pattern $\{(1, 1, 4), (1, 0, 0), (0, 1, 4)\}$, while the node x_1^2 and its only child x_1 determine the pattern $\{(2, 0, 0), (1, 0, 0)\}$. As a consequence of Proposition 19, we see that for three-element patterns P the respective moment body $\mathcal{M}_P(K)$ is a polytope. For the two-element patterns $P = \{(2, 0, 0), (1, 0, 0)\}$ and $P = \{(0, 0, 4), (0, 0, 1)\}$ the respective moment bodies $\mathcal{M}_P(K)$ are not polytopes, but they are two-dimensional bodies, for which an outer polyhedral approximation can be constructed rather easily. Hence, for this example, we establish a sparse polyhedral relaxation of (POP) using (P-RLX) with a pattern family arising from an expression tree.

There is yet another possibility. If we allow products with arbitrary many factors as elementary operations, then each monomial x^{α} can be written as a product of powers of the variables. For the above example, we obtain the expression tree



The difference in constructing patterns for this tree is that some of the nodes determine patterns of the form $\{\alpha_1 e_1, \dots, \alpha_n e_n, \alpha\}$. E.g., $x_1x_2x_3^4$ together with its children defines the pattern $\{(1, 0, 0), (0, 1, 0), (0, 0, 4), (1, 1, 4)\}$. For such patterns, Proposition 17 guarantees that the respective moment body $\mathcal{M}_P(K)$ is a polytope. Hence, the comments for the previous expression tree still apply.

2.3 Sparsifying dense approaches

We want to carry over Lasserre’s semidefinite relaxations of the moment body $\mathcal{M}_{\mathbb{N}^n_d}(X)$ body to a relaxation of $\mathcal{M}_P(X)$ for some smaller sets P , which can be used as patterns in (P-RLX).

For a pattern $P \subseteq \mathbb{N}^n$, we already introduced the notation $x^P := (x^{\alpha})_{\alpha \in P}$. Given a matrix $\Gamma = (\gamma(1), \dots, \gamma(k)) \in \mathbb{N}^{n \times k}$, we use the similar notation with $x^{\Gamma} := (x^{\gamma(1)}, \dots, x^{\gamma(k)})$ along with the vectors $l_{\Gamma} := (l_{\gamma(1)}, \dots, l_{\gamma(k)})$ and

$u_\Gamma := (u_{\gamma(1)}, \dots, u_{\gamma(k)})$. For a set X , we introduce the set $X^\Gamma := \{x^\Gamma : x \in X\}$, which is the image of X under the map $x \mapsto x^\Gamma$.

Assuming that Γ has rank k , one can now define the pattern $\Gamma P := \{\Gamma\alpha : \alpha \in P\}$, which we call the image of P under Γ . The exponents in P are in one-to-one correspondence with the exponents in ΓP , since Γ has rank k . We now map the k -dimensional patterns \mathbb{N}_d^k to patterns $\Gamma\mathbb{N}_d^k$. Clearly, one can generate many copies of such patterns by considering many different choices of Γ , and then use such patterns within (P-RLX).

We call monomials $x^{\gamma(1)}, \dots, x^{\gamma(k)}$ variable-independent if no variable x_i occurs in more than one of these k monomials. This means the supports of the exponent vectors $\gamma(1), \dots, \gamma(k)$ are pairwise disjoint.

► **Proposition 20.** *Let $\Gamma = (\gamma(1), \dots, \gamma(k)) \in \mathbb{N}^{n \times k}$ be a rank k matrix and consider a pattern $P \subseteq \mathbb{N}^k$. Then for the moment body of the pattern ΓP , one has*

$$\mathcal{M}_{\Gamma P}(X) = \mathcal{M}_P(X^\Gamma).$$

If $X = K = \text{Box}(l, u)$ and if $x^{\gamma(1)}, \dots, x^{\gamma(k)}$ are variable-independent monomials, then $K^\Gamma = \text{Box}(l_\Gamma, u_\Gamma)$.

Proof. Since Γ has rank k , the map $\alpha \mapsto \Gamma\alpha$ from P to ΓP is bijective. The pattern P gives the system of monomials y^P in the variables $y = (y_1, \dots, y_k)$, whereas the pattern ΓP gives the system of monomials $x^{\Gamma P}$. One can obtain $x^{\Gamma P}$ by substituting $y = x^\Gamma$ into y^P . In the coordinate form, the substitution can be written as $y_i = x^{\gamma(i)}$. One has

$$\begin{aligned} \mathcal{M}_{\Gamma P}(X) &= \text{conv}\{(x^\Gamma)^\alpha : \alpha \in P, x \in X\} \\ &= \text{conv}\{(y^\alpha)_{\alpha \in P} : y \in X^\Gamma\} \\ &= \mathcal{M}_P(X^\Gamma). \end{aligned}$$

Assume now that $X = K = \text{Box}(l, u)$ and that the monomials $x^{\gamma(1)}, \dots, x^{\gamma(k)}$ are variable-independent. In this case, by letting x vary in $K = \text{Box}(l, u)$, we can fix the values of the k monomials $x^{\gamma(i)}$ independently. Since $x^{\gamma(i)}$ takes values in $[l_{\gamma(i)}, u_{\gamma(i)}]$, we obtain $K^\Gamma = \text{Box}(l_\Gamma, u_\Gamma)$. ◀

If $\Gamma = (\gamma(1), \dots, \gamma(k)) \in \mathbb{N}^{n \times k}$ is such that the monomials $x^{\gamma(1)}, \dots, x^{\gamma(k)}$ are variable-independent, then we call the pattern $\Gamma\mathbb{N}_d^k$ a truncated sub-monoid pattern. If $k = 1$, then we call $\Gamma\mathbb{N}_d^k$ a chain. Thus, a chain is a pattern of the form $\{0, \gamma, \dots, d\gamma\}$, where $\gamma \in \mathbb{N}^n$.

Applying Proposition 20 to the moment body $\mathcal{M}_{\Gamma\mathbb{N}_d^k}(X)$ of a truncated submonoid pattern, we can transfer the relaxation techniques for $\mathcal{M}_{\mathbb{N}_d^n}(X)$ to the body $\mathcal{M}_{\Gamma\mathbb{N}_d^k}(X)$. In principle, both bound-factor relaxations and Lasserre's semidefinite relaxation can be chosen. In Section C we shall use Lasserre's relaxation, thus we only spell out the resulting Lasserre-type moment relaxation of $\mathcal{M}_{\Gamma\mathbb{N}_{2d}^k}(X)$ (we replace d by $2d$, since we work with even degrees in Lasserre's approach). For box constraints we have $K^\Gamma = \text{Box}(l_\Gamma, u_\Gamma)$. The Lasserre-type relaxation of $\mathcal{M}_{\Gamma\mathbb{N}_{2d}^k}$ is the following set

$$\text{MR}_{\Gamma,d}(K) := \left\{ v \in \mathbb{R}^{\Gamma\mathbb{N}_{2d}^k} : L_v(M_{\Gamma\mathbb{N}_d^k}) \succeq 0 \text{ and } L_v(h_i M_{\Gamma\mathbb{N}_{d-1}^k}) \succeq 0 \text{ for all } i \in [k] \right\}$$

where $h_i := (x^{\gamma(i)} - l_{\gamma(i)})(u_{\gamma(i)} - x^{\gamma(i)})$. We call $\text{MR}_{\Gamma,d}(K)$ the semidefinite relaxation of $\mathcal{M}_{\Gamma\mathbb{N}_d^k}(K)$ and use it in the computations in the Appendix.

► **Example 21.** Consider the box $K = [-1, 1] \times [-2, 2]$. We choose $d = 1$ and $\Gamma = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$. Then $\Gamma\mathbb{N}_{2d}^2 = \Gamma\mathbb{N}_2^2 = \{(0, 0), (2, 0), (0, 2), (4, 0), (2, 2), (0, 4)\}$ is the pattern. The monomials x_1^2 and x_2^2 range in the segments $[0, 1]$ and $[0, 4]$ for $x \in K$. That is, $l_{2,0} = 0, u_{2,0} = 1, l_{0,2} = 0, u_{0,2} = 4$. Consequently, $\text{MR}_{\Gamma,d}(K)$ is given by the LMIs

$$\begin{aligned} L_v \left(\begin{pmatrix} 1 & x_1^2 & x_2^2 \\ x_1^2 & x_1^4 & x_1^2 x_2^2 \\ x_2^2 & x_1^2 x_2^2 & x_2^4 \end{pmatrix} \right) &:= \begin{pmatrix} 1 & v_{2,0} & v_{0,2} \\ v_{2,0} & v_{4,0} & v_{2,2} \\ v_{0,2} & v_{2,2} & v_{0,4} \end{pmatrix} \succeq 0, \\ L_v(x_1^2(1 - x_1^2)) &:= v_{2,0} - v_{4,0} \geq 0, \\ L_v(x_2^2(4 - x_2^2)) &:= 4v_{0,2} - v_{0,4} \geq 0. \end{aligned}$$

► **Example 22.** Consider the box $K = [-1, 2] \times [1, 2]$ and the chain pattern $P = \{0, \gamma, 2\gamma\}$ with $\gamma = (1, 2)$. The Lasserre-type relaxation for this choice is $\text{MR}_{\Gamma,d}(K)$ with $\Gamma = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $d = 1$. One has $x^\gamma = x_1 x_2^2$. When $x \in K$, one has $-1 \leq x_1 \leq 2$ and $1 \leq x_2^2 \leq 4$. It follows that the least value for $x^\gamma = x_1 x_2^2$ is attained for $x_1 = -1, x_2^2 = 4$

and the largest value for x^γ is attained for $x_1 = 2, x_2^2 = 4$. That is, $l_\gamma = -4$ and $u_\gamma = 8$. Hence, $\text{MR}_{\Gamma,d}(K)$ is given by the LMIs

$$L_v \left(\begin{pmatrix} 1 & x_1 x_2^2 \\ x_1 x_2^2 & x_1^2 x_2^4 \end{pmatrix} \right) := \begin{pmatrix} 1 & v_{1,2} \\ v_{1,2} & v_{2,4} \end{pmatrix} \succeq 0, \\ L_v((x_1 x_2^2 + 4)(8 - x_1 x_2^2)) := 32 + 4v_{1,2} + v_{2,4} \geq 0.$$

2.4 Shifting patterns

For all considered patterns we can also construct a shifted version, which may be useful in computations. Note that if a pattern P does not contain the zero exponent, we can pass from P to $P \cup \{0\}$ without any significant changes in the relaxation, because the constraint $v_{P \cup \{0\}} \in \mathcal{M}_{P \cup \{0\}}(X)$ is equivalent to $v_P \in \mathcal{M}_P(X)$, $v_0 = 1$.

► **Proposition 23.** *Consider a pattern $P \subseteq \mathbb{N}^n$ with $0 \in P$ and an exponent vector $\eta \in \mathbb{N}^n \setminus \{0\}$ such that the two monomials x^η and x^α are variable-independent for each choice of $\alpha \in P$. Then the pattern $\eta + P$ is given by*

$$\mathcal{M}_{\eta+P}(X) = \{(v_{\eta+\beta})_{\beta \in P} \in \text{cone } \mathcal{M}_P(X) : l_\eta \leq v_\eta \leq u_\eta\}.$$

Proof. To clarify the asserted equality, we note that although $\text{cone } \mathcal{M}_P(X)$ is a subset of \mathbb{R}^P and $\mathcal{M}_{\eta+P}(X)$ is a subset of $\mathbb{R}^{\eta+P}$, we can identify \mathbb{R}^P and $\mathbb{R}^{\eta+P}$ in view of the bijection $\beta \leftrightarrow \eta + \beta$ between the index sets P and $\eta + P$. Due to variable-independence of the monomials x^η and x^α , the choice of values of x^η is independent on the choice of $m(x) := (x^\alpha)_{\alpha \in P}$, as x varies in X . Hence

$$\begin{aligned} \mathcal{M}_{\eta+P}(X) &= \text{conv}\{t x^\alpha : t \in [l_\eta, u_\eta], x \in X\} \\ &= \text{conv}\{t u : t \in [l_\eta, u_\eta], u \in \mathcal{M}_P(X)\}, \\ &= \text{conv}\{t u : t \in \{l_\eta, u_\eta\}, u \in \mathcal{M}_P(X)\}. \end{aligned}$$

Thus, $\mathcal{M}_{\eta+P}(X)$ is the convex hull of two parallel cross-sections of $\text{cone } \mathcal{M}_P(X)$ at “heights” l_η and u_η , respectively. This readily gives the assertion. ◀

Proposition 23 gives a way to establish formulations of shifted patterns, in particular, shifted truncated sub-monoids and shifted chains. Since we employ the conic-hull operation, we need to homogenize constraints. Both linear and semidefinite constraints can be homogenized. We give a simple example, illustrating how shifting works with respect to linear constraints.

► **Example 24.** Consider the pattern $\{0, 1\}^2 \times \{1\}$, which is a shift of $P = \{0, 1\}^2 \times \{1\}$ by the vector $\eta = (0, 0, 1)$ and let us choose $X = [0, 1]^2 \times [1, 2]$. The moment body $\mathcal{M}_P(X)$ is a polytope defined by the McCormick inequalities $L_v(x_1 x_2) := v_{1,1,0} \geq 0$, $L_v(x_1(1 - x_2)) := v_{1,0,0} - v_{1,1,0} \geq 0$, $L_v((1 - x_1)x_2) := v_{0,1,0} - v_{1,1,0} \geq 0$, $L_v((1 - x_1)(1 - x_2)) := 1 - v_{1,0,0} - v_{0,1,0} + v_{1,1,0} \geq 0$. When we homogenize, the 1 in the last inequality gets replaced by $v_{0,0,0}$. Finally, we shift the indices by adding $\eta = (0, 0, 1)$ and add the constraint $1 \leq v_{0,0,1} \leq 2$, resulting in the linear inequalities

$$\begin{aligned} v_{1,1,1} &\geq 0, \\ v_{1,0,1} - v_{1,1,1} &\geq 0, \\ v_{0,1,1} - v_{1,1,1} &\geq 0, \\ v_{0,0,1} - v_{1,0,1} - v_{0,1,1} + v_{1,1,1} &\geq 0, \\ 1 &\leq v_{0,0,1} \leq 2 \end{aligned}$$

that define $\mathcal{M}_{\{0,1\}^2 \times \{1\}}([0, 1]^2 \times [1, 2])$.

► **Example 25.** We give a small example that shows how to derive a semidefinite formulation of a shifted chain. Consider the shifted chain $\eta + P = \{(0, 1), (1, 1), (2, 1), (3, 1), (4, 1)\}$ with $P = \{(0, 0), (1, 0), (2, 0), (3, 0), (4, 0)\}$ and $\eta = (0, 1)$. We choose $X = [-1, 1] \times [1, 2]$. The moment body $\mathcal{M}_P(X)$ is described by the two LMIs

$$L_v \left(\begin{pmatrix} 1 & x_1 & x_1^2 \\ x_1 & x_1^2 & x_1^3 \\ x_1^2 & x_1^3 & x_1^4 \end{pmatrix} \right) := \begin{pmatrix} 1 & v_{1,0} & v_{2,0} \\ v_{1,0} & v_{2,0} & v_{3,0} \\ v_{2,0} & v_{3,0} & v_{4,0} \end{pmatrix} \succeq 0, \\ L_v \left((1 - x_1^2) \begin{pmatrix} 1 & x_1 \\ x_1 & x_1^2 \end{pmatrix} \right) := \begin{pmatrix} 1 - v_{2,0} & v_{1,0} - v_{3,0} \\ v_{1,0} - v_{3,0} & v_{2,0} - v_{4,0} \end{pmatrix} \succeq 0$$

Replacing 1 by $v_{0,0}$, then shifting the indices by $(0, 1)$ and introducing the constraint $1 \leq v_{0,1} \leq 2$, we arrive at the following description of $\mathcal{M}_{\eta+P}(X)$:

$$\begin{aligned} \begin{pmatrix} v_{0,1} & v_{1,1} & v_{2,1} \\ v_{1,1} & v_{2,1} & v_{3,1} \\ v_{2,1} & v_{3,1} & v_{4,1} \end{pmatrix} \succeq 0, \\ \begin{pmatrix} v_{0,1} - v_{2,1} & v_{1,1} - v_{3,1} \\ v_{1,1} - v_{3,1} & v_{2,1} - v_{4,1} \end{pmatrix} \succeq 0, \\ 1 \leq v_{0,1} \leq 2. \end{aligned}$$

2.5 Summary and Visualization of Patterns

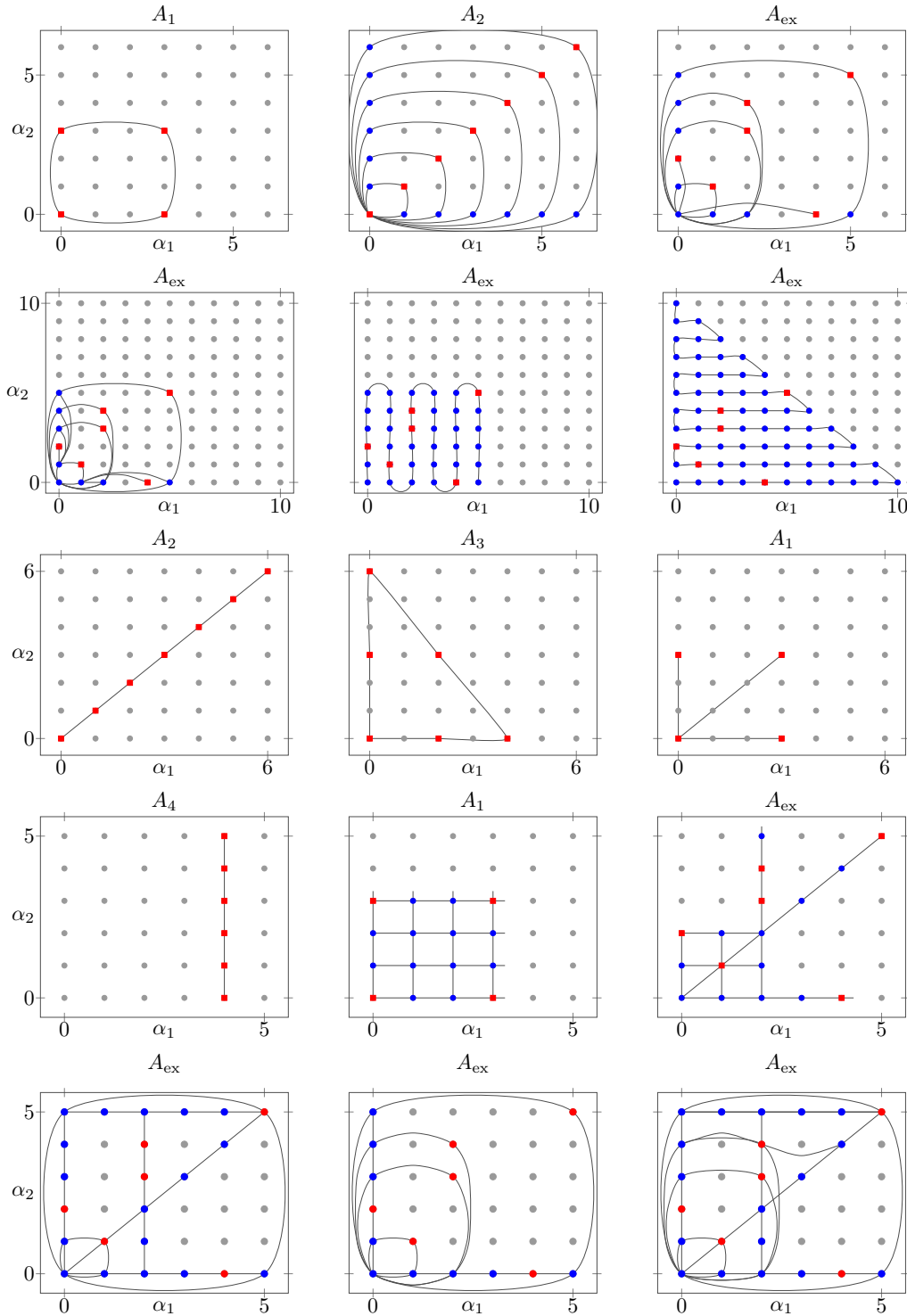
We have seen a variety of different patterns that can be used in our general convexification and sparsification framework. It is possible to combine the above approaches to obtain a pattern family F . This can be done problem-specifically, by exploiting properties of the underlying exponent set A to obtain a beneficial relation between computational cost and tightness of the relaxation. Examples of such customized pattern families combining multilinear patterns and shifted chains are visualized below and evaluated computationally in Section C.3.4.

One additional advantage of the abstracted concept of monomial patterns is that it allows a general way of visualizing which monomials and which auxiliary lifted variables are involved.

► **Example 26.** We consider different exponent sets for $n = 2$ for visualization,

$$\begin{aligned} A_1 &:= \{(0, 0), (0, 3), (3, 0), (3, 3)\}, \\ A_2 &:= \{(0, 0), (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)\}, \\ A_3 &:= \{(0, 0), (0, 3), (0, 6), (2, 0), (2, 3), (4, 0)\}, \\ A_4 &:= \{(4, 0), (4, 1), (4, 2), (4, 3), (4, 4), (4, 5)\}, \\ A_{\text{ex}} &:= \{(0, 2), (1, 1), (2, 3), (2, 4), (4, 0), (5, 5)\}. \end{aligned}$$

Figure 2 shows a visualization for them and some of the patterns discussed above. In addition, three custom pattern families F^1 , F^2 , and F^3 based on combinations of multilinear patterns, chains, and shifted chains applied to A_{ex} are also visualized in the last row of Figure 2. They will be used for numerical results in Section C.3.4.



■ **Figure 2** Visualization of involved variables v_α in exemplary patterns. The title of each subplot refers to the corresponding set A from Example 26. Exponents $\alpha \in A \subset \mathbb{N}^2$ are depicted as red squares. Pattern-specific auxiliary exponents $\alpha \in \cup_i P_i$ are depicted as blue dots. Curves connect all exponents $\alpha \in P_i$. As a rule of thumb, the largest cardinality $|P_i|$ has a strong influence on the overall runtime to solve (P-RLX). **First row:** multilinear patterns with $P = \{0, 1\}^2$, Section 2.2.2 **Second row:** expression tree, Section 2.2.2, bound-factor product, Section 2.2.2, and moment relaxation, Section 2.1 **Third row:** different truncated submonoids, Section 2.3 **Fourth row:** different shifted chains, Section 2.4 **Fifth row:** Families F^1, F^2, F^3 combining multilinear and shifted chains patterns

3 Conic pattern relaxations and their duality

We now focus on a different aspect of polynomial optimization – duality. Complementary, but also based on the concept of sparsification, we shall see how patterns corresponding to some popular dual approaches in polynomial optimization arise from a unifying, general framework.

3.1 Conic pattern relaxations

We introduce a conic analog of (P-RLX). To this end, analogously to how we defined the moment body $\mathcal{M}_A(X)$ using the convex hull, we introduce the *moment cone* $\mathcal{C}_A(X) := \overline{\text{cone}\{x^A : x \in X\}}$, defined using the conic hull. In the conic setting, we need to change our treatment of v_0 . In the discussion below, v_0 is a variable and not the constant 1. In particular, the use of the functional L_v is modified accordingly.

We give an example in which the relation between the moment body and the moment cone is straightforward and another example, where the relation is intricate. The two situations differ by the condition whether the set of exponent vectors contains the zero exponent.

► **Example 27.** Let $X = \mathbb{R}$. For $A = \{0, 1, 2\}$, one has

$$\mathcal{C}_A(X) = \left\{ \begin{pmatrix} v_0 \\ v_1 \\ v_2 \end{pmatrix} \in \mathbb{R}^A : \begin{pmatrix} v_0 & v_1 \\ v_1 & v_2 \end{pmatrix} \succeq 0 \right\},$$

while

$$\mathcal{M}_A(X) = \left\{ \begin{pmatrix} 1 \\ v_1 \\ v_2 \end{pmatrix} \in \mathbb{R}^A : \begin{pmatrix} 1 & v_1 \\ v_1 & v_2 \end{pmatrix} \succeq 0 \right\}$$

is the cross-section of $\mathcal{C}_A(X)$ by the plane $v_0 = 1$. So, here $\mathcal{C}_A(X)$ is a homogenization of $\mathcal{M}_A(X)$ and $\mathcal{M}_A(X)$ is a de-homogenization of $\mathcal{C}_A(X)$. That is, the relation between $\mathcal{C}_A(X)$ and $\mathcal{M}_A(X)$ is straightforward.

► **Example 28.** Consider the case $A = \{2, 3, 4\}$ and $X = \mathbb{R}$. Obviously, a polynomial $f(x) \in \mathbb{R}[x]_A$ is non-negative if and only if $f(x)/x^2 \in \mathbb{R}[x]_{\{0,1,2\}}$ is non-negative. This implies that non-negative linear functionals on $\mathcal{C}_A(X)$ correspond to non-negative linear functionals on $\mathcal{C}_{\{0,1,2\}}(X)$. Thus, by the separation theorem for convex cones, $\mathcal{C}_A(X)$ is a copy of $\mathcal{C}_{\{0,1,2\}}(X)$ so that one has

$$\mathcal{C}_A(X) = \left\{ \begin{pmatrix} v_2 \\ v_3 \\ v_4 \end{pmatrix} \in \mathbb{R}^A : \begin{pmatrix} v_2 & v_3 \\ v_3 & v_4 \end{pmatrix} \succeq 0 \right\}$$

The moment body $\mathcal{M}_A(X)$ can be obtained by taking the cross-section of $\mathcal{C}_{\{0\} \cup A}(X)$ by the hyperplane $v_0 = 1$. By standard results on the moment problem for the real line \mathbb{R} , we know that the moment cone $\mathcal{C}_{\{0,1,2,3,4\}}(X)$ is defined by $L_v(M_{\{0,1,2\}}) \succeq 0$. Consequently, since $\mathcal{C}_{\{0\} \cup A}$ is a projection of $\mathcal{C}_{\{0,1,2,3,4\}}(X)$, we have

$$\mathcal{C}_{\{0\} \cup A}(X) = \left\{ \begin{pmatrix} v_0 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} \in \mathbb{R}^{\{0\} \cup A} : \underbrace{\begin{pmatrix} v_0 & v_1 & v_2 \\ v_1 & v_2 & v_3 \\ v_2 & v_3 & v_4 \end{pmatrix}}_{=: L_v(M_{\{0,1,2\}})} \succeq 0 \text{ for some } v_1 \in \mathbb{R} \right\}$$

The semidefinite condition defining $\mathcal{C}_{\{0\} \cup A}(X)$ is an instance of a positive semidefinite matrix-completion problem for a chordal graph. We refer to Chapter 10 of [77] for details. Hence, by [77, Theorem 10.1], this semidefinite condition can be phrased via the positive semidefiniteness of the sub-matrices in the matrix defining $\mathcal{C}_{\{0\} \cup A}(X)$ that do not involve v_1 . Consequently,

$$\mathcal{C}_{\{0\} \cup A}(X) = \left\{ \begin{pmatrix} v_0 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} \in \mathbb{R}^{\{0\} \cup A} : \begin{pmatrix} v_0 & v_2 \\ v_2 & v_4 \end{pmatrix} \succeq 0, \begin{pmatrix} v_2 & v_3 \\ v_3 & v_4 \end{pmatrix} \succeq 0 \right\}$$

By fixing $v_0 = 1$ we obtain a description of $\mathcal{M}_A(X)$ by two LMIs with matrices of size two:

$$\mathcal{M}_A(X) = \left\{ \begin{pmatrix} v_2 \\ v_3 \\ v_4 \end{pmatrix} \in \mathbb{R}^A : \begin{pmatrix} 1 & v_2 \\ v_2 & v_4 \end{pmatrix} \succeq 0, \begin{pmatrix} v_2 & v_3 \\ v_3 & v_4 \end{pmatrix} \succeq 0 \right\}$$

We thus see that the description of $\mathcal{M}_A(X)$ contains an LMI not present in the description of $\mathcal{C}_A(X)$.

For a finite exponent set A with $0 \in A$, we formulate the conic convexification of the problem $\inf_{x \in X} f(x)$ of optimizing a polynomial $f \in \mathbb{R}[x]_A$ over a set $X \subseteq \mathbb{R}^n$, as the problem $\inf\{L_v(f) : v \in \mathbb{R}^A, v_0 = 1, v \in \mathcal{C}_A(X)\}$. This is nothing but (CVX-POP), since the constraints $v_0 = 1, v \in \mathcal{C}_A(X)$ are nothing but the constraint $v \in \mathcal{M}_A(X)$. But when we move on to a sparse relaxation, the conic sparse relaxation differs from (P-RLX) in general. As with (P-RLX), we fix a pattern family $\mathcal{F} = \{P_1, \dots, P_N\}$ such that $B := P_1 \cup \dots \cup P_N$ contains A as a subset. We call

$$\inf\{L_v(f) : v_0 = 1, v_{P_i} \in \mathcal{C}_{P_i}(X) \text{ for all } i \in [N]\} \quad (\text{C-P-RLX})$$

the *conic pattern relaxation* of the problem $\inf_{x \in X} f(x)$ for the pattern family $\mathcal{F} = \{P_1, \dots, P_N\}$. When all P_i contain 0, (C-P-RLX) is nothing but (P-RLX), but when some of the P_i 's do not contain 0, (C-P-RLX) is a coarser convexification than (P-RLX) in general.

Example 29 illustrates the difference between (P-RLX) and (C-P-RLX).

► **Example 29.** Consider $A = \{0, 1, 2, 3, 4\}$, $X = \mathbb{R}$ and the pattern family $\mathcal{F} = \{P_1, P_2\}$ with $P_1 = \{0, 1, 2\}$ and $P_2 = \{2, 3, 4\}$. For this choice, (P-RLX), formulated as

$$\inf \left\{ \sum_{i=0}^4 f_i v_i : (v_0, v_1, v_2) \in \mathcal{M}_{P_1}(X), (v_2, v_3, v_4) \in \mathcal{M}_{P_2}(X) \right\},$$

gives the exact optimal value of $\inf_{x \in X} f(x)$ when $f \in \mathbb{R}[x]_A$ satisfies $f_0 = f_1 = f_2 = 0$. In contrast to this, (C-P-RLX), formulated as

$$\inf \left\{ \sum_{i=0}^4 f_i v_i : v_0 = 1, (v_0, v_1, v_2) \in \mathcal{C}_{P_1}(X), (v_2, v_3, v_4) \in \mathcal{C}_{P_2}(X) \right\},$$

gives the optimal value $-\infty$ or 0 when $f_0 = f_1 = f_2 = 0$. To see this, note that $(v_2, v_3, v_4) \in \mathcal{C}_{P_2}(X)$ is a positively homogeneous constraint so that validity implies the validity of $(\lambda v_2, \lambda v_3, \lambda v_4) \in \mathcal{C}_{P_2}(X)$ for each $\lambda \in \mathbb{R}_+$. Consequently, $\inf\{(f_2(\lambda v_2) + f_3(\lambda v_3) + f_4(\lambda v_4)) : \lambda \in \mathbb{R}_+\}$ is $-\infty$ or 0, depending on whether $f_2 v_2 + f_3 v_3 + f_4 v_4$ is negative or not.

(C-P-RLX) is a coarser convexification of $\inf_{x \in X} f(x)$ than (P-RLX), but due to the direct presence of cones in (C-P-RLX), it has an advantage of being directly connected with conic optimization. In addition, the cones $\mathcal{C}_{P_i}(X)$ involved in (C-P-RLX) might have a simpler description than the respective moment bodies. We introduce the notation

$$\text{CRLX}_{\mathcal{F}}(X) := \{v \in \mathbb{R}^B : v_{P_i} \in \mathcal{C}_{P_i}(X) \text{ for all } i \in [N]\},$$

for the cone defining the conic moment relaxation. This cone is a relaxation of the cone $\mathcal{C}_A(X)$ in the sense of the inclusion $\mathcal{C}_A(X) \subseteq \text{CRLX}_{\mathcal{F}}(X)_A$. The problem (C-P-RLX) can be written as $\inf\{L_v(f) : v_0 = 1, v \in \text{CRLX}_{\mathcal{F}}(X)\}$.

3.2 Conic optimization and conic duality

We provide a brief overview of conic optimization and conic duality, which we present in a form tailored to our purposes. Dual cones can be formulated for general Euclidean spaces. Here we use a definition directly addressing the real vector space and standard scalar product.

► **Definition 30.** The dual cone of a set $C \subseteq \mathbb{R}^m$ is the closed convex cone

$$C^* := \{y \in \mathbb{R}^m : \langle y, x \rangle \geq 0 \text{ for all } x \in C\},$$

where $\langle x, y \rangle := x^\top y$ is the standard scalar product of x and y .

By separation theorems, for every closed convex cone C , one has $C^{**} = C$.

Conic programming is a general template of optimization problems with a linear objective function and so-called conic constraints. We define a *conic problem* as a problem of the form

$$\inf\{\langle c, x \rangle : x \in C, b - Ax \in D\} \quad (\text{CP})$$

with the linear objective function $x \in \mathbb{R}^n \mapsto \langle c, x \rangle$ given by $c \in \mathbb{R}^n$, the conic constraint $x \in C$ on the vector x of decision variables, defined using a closed convex cone $C \subseteq \mathbb{R}^n$, and a linear conic constraint $b - Ax \in D$, defined using a vector $b \in \mathbb{R}^m$, a matrix $A \in \mathbb{R}^{m \times n}$ and a closed convex cone $D \subseteq \mathbb{R}^m$. If we choose $D = \{0\}$, the constraint $b - Ax \in D$ amounts to the system of linear equations $Ax = b$. If we choose $D = \mathbb{R}_+^m$, the constraint $b - Ax \in D$ becomes a system of linear inequalities $Ax \leq b$. If we choose $C = \mathbb{R}_+^n$, the constraint $x \in C$ is the non-negativity constraint on the x variables. These observations shows that linear programming in its basic versions is captured as a special case of (CP). The dual problem of (CP) is defined as

$$\sup\{\langle b, y \rangle : c - A^\top y \in C^*, y \in D^*\} \quad (\text{CP-D})$$

Rather than presenting a general conic-duality theorem relating (CP) and (CP-D), we focus on its special case, which we need for applications in polynomial optimization.

► **Proposition 31.** *Let $C \subseteq \mathbb{R}^m$ be a pointed closed convex cone, let $p \in \mathbb{R}^m$ and $q \in C \setminus \{0\}$. Then*

$$\sup\{\lambda \in \mathbb{R} : p - \lambda q \in C\} = \inf\{\langle p, v \rangle : v \in C^*, \langle q, v \rangle = 1\}. \quad (15)$$

The proposition is borrowed from [9, Proposition 2.4]. Its proof is an easy exercise in convexity theory.

► **Remark 32.** (15) asserts the equivalence of (CP) and (CP-D) for a special choice of A, b, c and the cones C and D .

Let us also establish an abstract “sparse” version of Proposition 31.

► **Proposition 33.** *Let $C_1, \dots, C_N \subseteq \mathbb{R}^m$ be closed convex cones, which are subsets of a pointed closed convex cone C , let $p \in \mathbb{R}^m$ and $q \in (C_1 + \dots + C_N) \setminus \{0\}$. Then*

$$\sup\{\lambda \in \mathbb{R} : p - \lambda q \in C_1 + \dots + C_N\} = \inf\{\langle p, v \rangle : v \in C_1^* \cap \dots \cap C_N^*, \langle q, v \rangle = 1\}. \quad (16)$$

Proof. Since $C_1 + \dots + C_N \subseteq C$ and C is a pointed closed convex cone, it follows that $C_1 + \dots + C_N$, too, is a pointed closed convex cone. Hence, the assertion follows by using Proposition 31 for the cone $C_1 + \dots + C_N$ in combination with the well-known equality $(C_1 + \dots + C_N)^* = C_1^* \cap \dots \cap C_N^*$. ◀

3.3 Duality for conic pattern relaxations

We apply conic optimization and duality as laid out in Section 3.2 to conic pattern relaxations. Let $\mathcal{P}_A(X)$ denote the set of polynomials $f \in \mathbb{R}[x]_A$ that are non-negative on X . It is clear that $\mathcal{P}_A(X)$ is a closed convex cone. Furthermore, if X has non-empty interior, the cone $\mathcal{P}_A(X)$ is pointed. Indeed, if $\mathcal{P}_A(X) \cap (-\mathcal{P}_A(X))$, as the set of polynomials $f \in \mathbb{R}[x]_A$ that are zero on X , consists only of the zero polynomial.

► **Remark 34.** The set $\mathbb{R}[x]_A$ can be identified with \mathbb{R}^A via the natural bijection $\sum_{\alpha \in A} f_\alpha x^\alpha \leftrightarrow (f_\alpha)_{\alpha \in A}$. With this identification, $L_v(f)$ is interpreted as the scalar product of $f \in \mathbb{R}[x]_A \leftrightarrow \mathbb{R}^A$ and $v \in \mathbb{R}^A$. Consequently, for $C \subseteq \mathbb{R}[x]_A$ we have $C^* = \{v \in \mathbb{R}^A : L_v(f) \geq 0 \text{ for all } f \in C\}$. Analogously, for $D \subseteq \mathbb{R}^A$, we can interpret D^* as a cone in $\mathbb{R}[x]_A$ and use the equality $D^* = \{f \in \mathbb{R}[x]_A : L_v(f) \geq 0 \text{ for all } v \in D\}$.

► **Proposition 35.** *Let $A \subseteq \mathbb{N}^n$ be a finite exponent set and $X \subseteq \mathbb{R}^n$. Then $\mathcal{C}_A(X)^* = \mathcal{M}_A(X)^* = \mathcal{P}_A(X)$.*

Proof. We have

$$\begin{aligned}
\mathcal{C}_A(X)^* &= \left\{ f \in \mathbb{R}[x]_A : \sum_{\alpha \in A} f_\alpha v_\alpha \geq 0 \text{ for all } v \in \mathcal{C}_A(X) \right\} \\
&= \left\{ f \in \mathbb{R}[x]_A : \sum_{\alpha \in A} f_\alpha v_\alpha \geq 0 \text{ for all } v \in \overline{\text{cone}\{x^A : x \in X\}} \right\} \\
&= \left\{ f \in \mathbb{R}[x]_A : \sum_{\alpha \in A} f_\alpha v_\alpha \geq 0 \text{ for all } v \in \{x^A : x \in X\} \right\} \\
&= \left\{ f \in \mathbb{R}[x]_A : \sum_{\alpha \in A} f_\alpha x^\alpha \geq 0 \text{ for all } x \in X \right\} \\
&= \mathcal{P}_A(X).
\end{aligned}$$

Analogously, we obtain $\mathcal{M}_A(X)^* = \mathcal{P}_A(X)$. ◀

The duality $\mathcal{M}_A(X)^* = \mathcal{P}_A(X)$ corresponds to a pair of optimization problems, dual to each other.

► **Proposition 36.** *Let $A \subseteq \mathbb{N}^n$ with $0 \in A$ and $f \in \mathbb{R}[x]_A$. Then*

$$\begin{aligned}
\sup\{\lambda \in \mathbb{R} : f - \lambda \in \mathcal{P}_A(X)\} &= \inf\{L_v(f) : v \in \mathcal{C}_A(X), v_0 = 1\} \\
&= \inf\{L_v(f) : v \in \mathcal{M}_A(X)\}.
\end{aligned}$$

Proof. The assertion is trivial, but it also follows from Proposition 31. ◀

In Proposition 36, the formal supremum-problem is a declaration of the intention to find lower bounds on f , while the infimum-problem (written in two equivalent versions) is a declaration of the intention to convexify. Proposition 36 serves as a starting point for establishing an analogous sparse version of duality.

► **Proposition 37.** *Let $P_1, \dots, P_N, A \subseteq \mathbb{N}^n$ be finite sets of exponent vectors satisfying $\{0\} \cup A \subseteq B := P_1 \cup \dots \cup P_N$. Let $f \in \mathbb{R}[x]_A$ and let $X \subseteq \mathbb{R}^n$ have non-empty interior. Then*

$$\begin{aligned}
\sup\{\lambda \in \mathbb{R} : f - \lambda \in \mathcal{P}_{P_1}(X) + \dots + \mathcal{P}_{P_N}(X)\} \\
= \inf\{L_v(f) : v_0 = 1, v_{P_i} \in \mathcal{C}_{P_i}(X) \text{ for every } i \in [N]\}.
\end{aligned}$$

Furthermore, if $0 \in P_i$ for every $i \in [N]$, then the infimum is equal to

$$\inf\{L_v(f) : v_{P_i} \in \mathcal{M}_{P_i}(X) \text{ for every } i \in [N]\}.$$

If, additionally, the set X is compact, then sup and inf are attained and can be replaced by max and min, respectively.

Proof. The cones $\mathcal{P}_{P_i}(X)$ are subsets of the closed pointed convex cone $\mathcal{P}_B(X)$, which shows that the assumptions of Proposition 33 are fulfilled for the cones $C_i = \mathcal{P}_{P_i}(X)$. As a direct consequence of Proposition 35, we obtain

$$(\mathcal{C}_A(X) \times \mathbb{R}^{B \setminus A})^* = (\mathcal{M}_A(X) \times \mathbb{R}^{B \setminus A})^* = \mathcal{P}_A(X) \subseteq \mathbb{R}[x]_B$$

which, inserted in Proposition 33, yields the main assertion. Now, assume that $0 \in P_i$ for every $i \in [N]$. Then one has $\mathcal{M}_{P_i}(X) = \{v \in \mathcal{C}_{P_i}(X) : v_0 = 1\}$. Assume additionally that X is compact. The infimum is finite and a minimum, because $\text{RLX}_{\mathcal{F}}(X) = \{v \in \mathbb{R}^B : v_{P_i} \in \mathcal{M}_{P_i}(X) \text{ for all } i \in [N]\}$ is a compact set contained in $\text{Box}(l_B, u_B)$. Hence the supremum is finite as well. To see that the (finite) supremum is attained it suffices to observe that $\mathcal{P}_{P_1}(X) + \dots + \mathcal{P}_{P_N}(X)$ is a closed convex cone so that $\{f - \lambda : \lambda \in \mathbb{R}, f - \lambda \in \mathcal{P}_{P_1}(X) + \dots + \mathcal{P}_{P_N}(X)\}$ is a ray with the end point of the ray corresponding to the λ , for which the supremum is attained. ◀

We call

$$\max\{\lambda \in \mathbb{R} : f - \lambda \in \mathcal{P}_{P_1}(X) + \dots + \mathcal{P}_{P_N}(X)\}. \tag{D-P-RLX}$$

the *dual pattern relaxation* for the pattern family $\mathcal{F} = \{P_1, \dots, P_N\}$. The constraint in (D-P-RLX) is a formal constraint $f - \lambda = g_1 + \dots + g_N$, involving the real-valued variable λ , certifying the bound $f \geq \lambda$ on f through

the ‘‘conic variables’’ $g_i \in \mathcal{P}_{P_i}(X)$. Analogously to our discussion of the primal problem (P-RLX), the conditions $g_i \in \mathcal{P}_{P_i}(X)$ are formal constraints, since (D-P-RLX) gives no clues on how to model these constraints. It is just a declaration that if the P_i ’s allow a sparse formulation of $g_i \in \mathcal{P}_{P_i}(X)$ in some optimization paradigm this implies the same for the model (D-P-RLX). For being able to use (D-P-RLX), one either needs to provide exact descriptions of $\mathcal{P}_{P_i}(X)$, which may be a description in the original space or a lifted description, or to model an *inner* approximation of $\mathcal{P}_{P_i}(X)$. We stress that relaxing (D-P-RLX) means finding formulations of inner approximations of $\mathcal{P}_{P_i}(X)$. Indeed, (D-P-RLX) is a maximization problem, so that a strengthening of the constraints $v_{P_i} \in \mathcal{P}_{P_i}(X)$ by replacing $\mathcal{P}_{P_i}(X)$ with smaller cones makes the maximum over λ ’s smaller. Thus, relaxing (D-P-RLX) means to determine a cone of particular types of non-negative polynomials with the support in P_i and use such polynomials for certifying bounds $f(x) \geq \lambda$ valid for $x \in X$.

3.4 Sparse conic approaches from the pattern perspective

One can develop a concrete sparse conic approach by starting with (C-P-RLX) and then deriving (D-P-RLX) via dualization, but one can also start with (D-P-RLX) and then derive (C-P-RLX) via dualization. If one starts with (C-P-RLX), one emphasizes convexification and then passes to (D-P-RLX), which is concerned with a certification of bounds using sparse non-negative polynomials. If one starts with (D-P-RLX) and then derives (C-P-RLX) as, e.g., in [34], one emphasizes certificates of bounds and then moves on to convexification. Emphasis on bound certification has been the favorite choice in the study of the SAGE, SOS, SDSOS, TS-SSOS and related approaches, which we are going to summarize below. The contents of the following subsections are mostly a survey of results about sparse approaches to polynomial optimization that are spread in the literature. Our intention was to provide a largely self-contained and unified presentation, in which specific approaches are derived as corollaries of general results. Our presentation of the duality for SONC and SAGE differs from the presentation of the duality in other sources such as [34]: rather than starting with a formulation of (D-P-RLX) and obtaining a formulation of (C-P-RLX) via the dualization, we start with a formulation of (C-P-RLX) and dualize it to a formulation of (D-P-RLX).

3.4.1 SAGE, SONC, and SDSOS

SAGE and SONC are closely related to each other [15, 34]. The patterns in both these approaches are simplicial circuits, but SAGE deals with the non-negativity on \mathbb{R}_+^n , while SONC with the non-negativity on \mathbb{R}^n . We start with a discussion of SAGE.

► **Proposition 38.** *Let $\gamma(0), \dots, \gamma(k) \in \mathbb{N}^n$ be affinely independent exponent vectors and let $\beta \in \mathbb{N}^n$ be an exponent vector in the relative interior of the k -dimensional simplex $\text{conv}(\gamma(0), \dots, \gamma(k))$. Consider the coefficients $\lambda_i > 0$ of the convex combination*

$$\begin{aligned} \beta &= \sum_{i=0}^k \lambda_i \gamma(i), \\ 1 &= \sum_{i=0}^k \lambda_i \end{aligned}$$

Then, for the pattern $P = \{\beta, \gamma(0), \dots, \gamma(k)\}$, one has

$$\mathcal{C}_P(\mathbb{R}_+^n) = \left\{ v \in \mathbb{R}_+^P : 0 \leq v_\beta \leq \prod_{i=0}^k v_{\gamma(i)}^{\lambda_i} \right\} \quad (17)$$

Proof. We denote the right-hand side of (17) as C . It is well known that the weighted geometric mean $g : \mathbb{R}_+^{k+1} \rightarrow \mathbb{R}$

$$g(t_0, \dots, t_k) = \prod_{i=0}^k t_i^{\lambda_i}$$

is a concave positively homogeneous function. Thus, C is a closed convex cone defined as the region below the graph of g and above the graph of the zero function on \mathbb{R}_+^{k+1} . To see that $\mathcal{C}_P(\mathbb{R}_+^n) \subseteq C$ is true, it suffices to observe that for the choice $v_\beta = x^\beta$ and $v_{\gamma(i)} = x^{\gamma(i)}$ with $x \in \mathbb{R}_+^n$, the inequalities defining C are true, as

the upper bound on v_β is attained with equality and the lower bound on v_β is trivially fulfilled. To verify the converse inclusion $C \subseteq \mathcal{C}_P(\mathbb{R}_+^n)$, we start with the set

$$G = \left\{ v \in \mathbb{R}_{>0}^P : v_\beta = \prod_{i=0}^k v_{\gamma(i)}^{\lambda_i} \right\},$$

the relative interior of the graph of g . Pick an arbitrary $v \in G$, fix $y_i = \ln v_{\gamma(i)}$ and consider the matrix $\Gamma = (\gamma(0), \dots, \gamma(k))$. Let $\mathbf{1} = (1, \dots, 1)^\top \in \mathbb{R}^{k+1}$. The matrix $M = (\Gamma^\top \mathbf{1})$ has rank $k+1$. Hence, the linear system $Mu = y$ with the right-hand side $y = (y_0, \dots, y_k)^\top$ has a solution $u = (u_1, \dots, u_{n+1})^\top$. Let $w = (u_1, \dots, u_n)^\top$. One has $y = Mu = \Gamma^\top w + \mathbf{1}u_{n+1}$, which can be spelled out as $\langle \gamma(i), w \rangle + u_{n+1} = y_i$. Exponentiation of the latter equality gives $x^{\gamma(i)} e^{u_{n+1}} = v_{\gamma(i)}$ with $x = (e^{u_1}, \dots, e^{u_n})^\top \in \mathbb{R}_{>0}^n$. For this choice of x and $c = e^{u_{n+1}}$, we obtain $cx^\beta = \prod_{i=0}^k (cx^{\gamma(i)})^{\lambda_i} = \sum_{i=0}^k v_{\gamma(i)} = v_\beta$ and thus $cx^P = v$. We have realized $v \in G$ as a point in $\mathcal{C}_P(\mathbb{R}_+^n)$ up to a scaling factor $c > 0$. Since $\mathcal{P}_P(\mathbb{R}_+^n)$ is a cone, $v \in G$ itself is contained in $\mathcal{C}_P(\mathbb{R}_+^n)$. This shows that $G \subseteq \mathcal{C}_P(\mathbb{R}_+^n)$. But then the topological closure \bar{G} , which is the whole graph of g , is also a subset of $\mathcal{C}_P(\mathbb{R}_+^n)$. To conclude the proof it thus suffices to show that $\text{conv}(\bar{G}) = C$. It is clear that if $(v_{\gamma(0)}, \dots, v_{\gamma(k)})$ is a standard unit vector and $v_\beta = 0$, then v is in the graph \bar{G} of g . But then the graph $\mathbb{R}_+^{k+1} \times \{0\}$ of the zero function, defined on \mathbb{R}_+^k is in $\text{conv}(\bar{G})$. This shows that the graph \bar{G} of g and the graph of the zero function on \mathbb{R}_+^k are both subsets of C . But since every point $v \in C$ lies above the graph of the zero function and below the graph of g , we conclude that $C = \text{conv}(\bar{G})$. ◀

We call a pattern P as in (17) a *SAGE-circuit*. For a family $\mathcal{F} = \{P_1, \dots, P_N\}$ of SAGE circuits and the domain $X = \mathbb{R}_+^n$, we call the problem (C-P-RLX) the *SAGE moment convexification* of the problem $\inf_{x \in X} f(x)$.

For establishing the dual of the SAGE moment convexification, we need to dualize the cone $\mathcal{C}_P(\mathbb{R}_+^n)$ from (17).

► **Definition 39.** For $\lambda = (\lambda_0, \dots, \lambda_k) \in \mathbb{R}_{>0}^{k+1}$ with $1 = \sum_{i=0}^k \lambda_i$, we define the geometric-mean cone for λ by

$$\text{GMC}_\lambda := \left\{ (y, t_0, \dots, t_k) \in \mathbb{R}_+^{k+2} : 0 \leq y \leq \prod_{i=0}^k t_i^{\lambda_i} \right\}.$$

Note that the geometric-mean cone is defined slightly differently in MOSEK's documentation [50].

► **Proposition 40.** In the notation of Definition 39, one has

$$(\text{GMC}_\lambda)^* = \left\{ (z, s_0, \dots, s_k) \in \mathbb{R} \times \mathbb{R}_+^{k+1} : z + \sum_{i=0}^k \left(\frac{s_i}{\lambda_i} \right)^{\lambda_i} \geq 0 \right\}. \quad (18)$$

Proof. We denote GMC_λ as C and the right-hand side of (18) as D . The scalar product of $(y, t_0, \dots, t_k) \in C$ and $(z, s_0, \dots, s_k) \in D$ satisfies

$$\begin{aligned} zy + \sum_{i=0}^k s_i t_i &= zy + \sum_{i=0}^k \lambda_i \frac{s_i}{\lambda_i} t_i \\ &\geq zy + \prod_{i=0}^k \left(\frac{s_i}{\lambda_i} \right)^{\lambda_i} \prod_{i=0}^k t_i^{\lambda_i} && \text{(by the AM-GM inequality)} \\ &\geq \left(z + \prod_{i=0}^k \left(\frac{s_i}{\lambda_i} \right)^{\lambda_i} \right) y \\ &\geq 0. \end{aligned}$$

This shows that $D \subseteq C^*$. Assume now $C^* \not\subseteq D$. Then there exists $(z, s_0, \dots, s_k) \in C^*$ with $s_0, \dots, s_k \geq 0$ and $z + \prod_{i=0}^k \left(\frac{s_i}{\lambda_i} \right)^{\lambda_i} < 0$. The continuity of this function allows finding slightly increased values $s_0, \dots, s_k > 0$ still resulting in $z + \prod_{i=0}^k \left(\frac{s_i}{\lambda_i} \right)^{\lambda_i} < 0$. Then, fixing $(y, t_0, \dots, t_k) \in C$ with $t_i = \frac{\lambda_i}{s_i}$ and $y = \prod_{i=0}^k t_i^{\lambda_i}$, for the scalar

product of (y, t_0, \dots, t_k) and (z, s_0, \dots, s_k) , we obtain

$$\begin{aligned} zy + \sum_{i=0}^k s_i t_i &= zy + \sum_{i=0}^k \lambda_i \\ &= zy + 1 \\ &< - \prod_{i=0}^k \left(\frac{s_i}{\lambda_i} \right)^{\lambda_i} y + 1 \\ &= - \prod_{i=0}^k \left(\frac{s_i}{\lambda_i} \right)^{\lambda_i} t_i^{\lambda_i} + 1 \\ &= 0. \end{aligned}$$

The scalar product is negative, which contradicts the fact that one of the vectors is in C^* and the other one is in C . This implies that $D = C^*$. ◀

As a direct consequence of Proposition 40, we obtain

► **Proposition 41.** *In the notation of Proposition 38, one has*

$$\mathcal{P}_P(\mathbb{R}_+^n) = \left\{ f \in \mathbb{R}[x]_P : f_\beta + \prod_{i=0}^k \left(\frac{f_{\gamma(i)}}{\lambda_i} \right)^{\lambda_i} \geq 0 \right\}.$$

Proof. The assertion follows from Propositions 38, 40, and the duality relation $\mathcal{C}_P(\mathbb{R}_+^n)^* = \mathcal{P}_P(\mathbb{R}_+^n)$. ◀

Proposition 41 yields an explicit formulation of the problem (D-P-RLX) on the domain $X = \mathbb{R}_+^n$ in the case when $\mathcal{F} = \{P_1, \dots, P_N\}$ is a family of SAGE-circuits. We call such a problem (D-P-RLX) the *SAGE relaxation* of $\inf_{x \in \mathbb{R}_+^n} f(x)$ with respect to the family of SAGE-circuits \mathcal{F} . The polynomials of from $\mathcal{P}_{P_1}(\mathbb{R}_+^n) + \dots + \mathcal{P}_{P_N}(\mathbb{R}_+^n)$, where P_1, \dots, P_N are SAGE-circuits, are called *SAGE polynomials*.

► **Remark 42.** Let us briefly discuss computational options for the SAGE moment convexification and the SAGE relaxation. The problems involve the cones GMC_λ and $(\text{GMC}_\lambda)^*$ for different choices of λ . The cones GMC_λ and $(\text{GMC}_\lambda)^*$ are similar in nature, because both involve the weighted geometric-mean in their description. It is known that GMC_λ and $(\text{GMC}_\lambda)^*$ are second-order cone representable [12]. The Mosek documentation [6] provides hints on how to model cones like GMC_λ and its dual. There are also further options, some of which rely on the substitution of variables $v_\alpha = e^{\nu_\alpha}$ for the SAGE moment convexification and $f_\alpha = e^{\phi_\alpha}$ for the SAGE relaxation.

We proceed with a similar derivation of primal and dual SONC relaxations.

► **Proposition 43.** *Let the assumptions of Proposition 38 hold and assume that additionally $\gamma(0), \dots, \gamma(k) \in 2\mathbb{N}^n$. Then, for $\beta \in 2\mathbb{N}^n$, one has*

$$\mathcal{C}_P(\mathbb{R}^n) = \left\{ v \in \mathbb{R}_+^P : 0 \leq v_\beta \leq \prod_{i=0}^k v_{\gamma(i)}^{\lambda_i} \right\} \quad (19)$$

and, for $\beta \notin 2\mathbb{N}^n$, one has

$$\mathcal{C}_P(\mathbb{R}^n) = \left\{ v \in \mathbb{R}_+^P : - \prod_{i=0}^k v_{\gamma(i)}^{\lambda_i} \leq v_\beta \leq \prod_{i=0}^k v_{\gamma(i)}^{\lambda_i} \right\} \quad (20)$$

Proof. If $\beta \in 2\mathbb{N}^n$, then $P \subseteq 2\mathbb{N}^n$. This implies that the value x^P does not depend on the signs of the coordinates x_1, \dots, x_n but only on the absolute values $|x_1|, \dots, |x_n|$. This implies $\mathcal{C}_P(\mathbb{R}^n) = \mathcal{C}_P(\mathbb{R}_+^n)$ so that the assertion follows from Proposition 38. If $\beta \notin 2\mathbb{N}^n$, then the sign of $x^{\gamma(i)}$ still does not depend on the signs of x_1, \dots, x_n but the sign of x^β depend on the signs of x_1, \dots, x_n and switching the sign of x_i , for which β_i is odd, generates a sign switch of x^β . This shows that $\mathcal{C}_P(\mathbb{R}^n)$ contains both $C' := \mathcal{C}_P(\mathbb{R}_+^n)$ and its reflection $C'' := \{v \in \mathbb{R}^P : (-v_\beta, v_{\gamma(0)}, \dots, v_{\gamma(k)}) \in \mathcal{C}_P(\mathbb{R}_+^n)\}$. Hence, $\mathcal{C}_P(\mathbb{R}^n) \supseteq C' \cup C''$, where, in view of Proposition 38, $C := C' \cup C''$ is the right-hand side of (20). To see that $\mathcal{C}_P(\mathbb{R}^n)$ is a subset of C it suffices to observe that, for every $x \in \mathbb{R}^n$, one has $x^P = (-|x^\beta|, x^{\gamma(0)}, \dots, x^{\gamma(k)})$ or $x^P = (|x^\beta|, x^{\gamma(0)}, \dots, x^{\gamma(k)})$, we see that for $v = x^P$, one of the two convex constraints that define C is attained with equality. This concludes the proof of the equality $\mathcal{C}_P(\mathbb{R}^n) = C$. ◀

We can dualize the cones from Proposition 43.

► **Proposition 44.** *In the notation of Proposition 43, for $\beta \in 2\mathbb{N}^n$, one has*

$$\mathcal{P}_P(\mathbb{R}^n) = \left\{ f \in \mathbb{R}[x]_P : f_\beta + \prod_{i=0}^k \left(\frac{f_{\gamma(i)}}{\lambda_i} \right)^{\lambda_i} \geq 0 \right\}$$

and, for $\beta \notin 2\mathbb{N}^n$, one has

$$\mathcal{P}_P(\mathbb{R}^n) = \left\{ f \in \mathbb{R}[x]_P : -|f_\beta| + \prod_{i=0}^k \left(\frac{f_{\gamma(i)}}{\lambda_i} \right)^{\lambda_i} \geq 0 \right\}.$$

Proof. If $\beta \in 2\mathbb{N}^n$, then we have $\mathcal{C}_P(\mathbb{R}^n) = \mathcal{C}_P(\mathbb{R}_+^n)$. Hence $\mathcal{P}_P(\mathbb{R}^n) = \mathcal{C}_P(\mathbb{R}^n)^* = \mathcal{C}_P(\mathbb{R}_+^n)^* = \mathcal{P}_P(\mathbb{R}_+^n)$, where $\mathcal{P}_P(\mathbb{R}_+^n)$ was determined in Proposition 41. Assume $\beta \notin 2\mathbb{N}^n$. We recall that $x^{\gamma(i)}$ does not depend on the signs of x_1, \dots, x_n because $\gamma(i) \in 2\mathbb{N}^n$, while x^β has a sign that can be switched by switching the sign of an x_i with odd β_i . This shows that $f_\beta x^\beta + \sum_{i=0}^k f_{\gamma(i)} x^{\gamma(i)}$ is non-negative on \mathbb{R}^n if and only if $-|f_\beta| x^\beta + \sum_{i=0}^k f_{\gamma(i)} x^{\gamma(i)}$ is non-negative on \mathbb{R}_+^n . ◀

If $\gamma(0), \dots, \gamma(k)$ are affinely independent exponent vectors that belong to $2\mathbb{N}^n$ and $\beta \in \mathbb{N}^n$ lies in the relative interior of the simplex with the vertices $\gamma(0), \dots, \gamma(k)$, we call $P = \{\beta, \gamma(0), \dots, \gamma(k)\}$ a *SONC circuit*. We call (C-P-RLX) on the domain $X = \mathbb{R}^n$ where $\mathcal{F} = \{P_1, \dots, P_N\}$ is a family of SONC circuits a *SONC moment convexification* of the unconstrained problem $\inf_{x \in \mathbb{R}^n} f(x)$, and we call the respective problem (D-P-RLX) a *SONC relaxation* of the unconstrained problem $\inf_{x \in \mathbb{R}^n} f(x)$. Polynomials from $\mathcal{P}_{P_1}(\mathbb{R}^n) + \dots + \mathcal{P}_{P_N}(\mathbb{R}^n)$, where P_1, \dots, P_N are SONC circuits, are called *SONC polynomials*.

From Proposition 44, we obtain an explicit description of the SONC relaxations for the problem $\inf_{x \in \mathbb{R}^n} f(x)$.

► **Remark 45.** Comments in Remark 42 regarding the computational options for SAGE relaxations also apply for SONC relaxations.

We call a pattern of the form $\{2\alpha, \alpha + \beta, 2\beta\}$ with $\alpha, \beta \in \mathbb{N}^n$ and $\alpha \neq \beta$ an SDSOS pattern. This is a special SONC pattern with $\alpha + \beta = \lambda_0(2\alpha) + \lambda_1(2\beta)$ and $\lambda_0 = \lambda_1 = \frac{1}{2}$.

► **Proposition 46.** *Let $P = \{2\alpha, \alpha + \beta, 2\beta\} \subseteq \mathbb{N}^n$ be an SDSOS pattern. Then, for $\alpha + \beta \in 2\mathbb{N}^n$, one has*

$$\begin{aligned} \mathcal{C}_P(\mathbb{R}^n) &= \{v \in \mathbb{R}_+^P : 0 \leq v_{\alpha+\beta}, v_{\alpha+\beta}^2 \leq v_{2\alpha} v_{2\beta}\}, \\ \mathcal{P}_P(\mathbb{R}^n) &= \{f \in \mathbb{R}[x]_P : f_{\alpha+\beta} + 2(f_{2\alpha} \cdot f_{2\beta})^{1/2} \geq 0\}. \end{aligned}$$

Furthermore, for $\alpha + \beta \notin 2\mathbb{N}^n$, one has

$$\begin{aligned} \mathcal{C}_P(\mathbb{R}^n) &= \{v \in \mathbb{R}_+^P : v_{\alpha+\beta}^2 \leq v_{2\alpha} v_{2\beta}\}, \\ \mathcal{P}_P(\mathbb{R}^n) &= \{v \in \mathbb{R}^P : f_{\alpha+\beta}^2 \leq 2f_{2\alpha} \cdot f_{2\beta}\}. \end{aligned}$$

Proof. The assertion follows from Propositions 43 and 44 applied to the special circuit P with $k = 1$ and $\lambda_0 = \lambda_1 = \frac{1}{2}$. ◀

Analogously to SONC relaxations, we define *SDSOS moment convexifications* and *SDSOS relaxations* of the problem $\inf_{x \in \mathbb{R}^n} f(x)$. If P_1, \dots, P_N are SDSOS patterns, then we call a polynomial from $\mathcal{P}_{P_1}(\mathbb{R}^n) + \dots + \mathcal{P}_{P_N}(\mathbb{R}^n)$ an *SDSOS polynomial*.

3.4.2 Sparse SOS certificates for unconstrained problems

Let \mathbb{S}_+^m be the cone of positive semidefinite matrices within the vector space \mathbb{S}^m of symmetric matrices of size m . We introduce the Euclidean structure on \mathbb{S}^m by endowing it with the trace scalar product $\langle U, V \rangle = \text{tr}(UV)$. The following is well known:

► **Proposition 47.** *The cone \mathbb{S}_+^m is self-dual. That is, $(\mathbb{S}_+^m)^* = \mathbb{S}_+^m$*

Proof. \mathbb{S}_+^m is the conic hull of the rank-one matrices uu^\top with $u \in \mathbb{R}^m$. Hence, $(\mathbb{S}_+^m)^*$ consists of matrices V that satisfy $\langle V, uu^\top \rangle \geq 0$ for every $u \in \mathbb{R}^m$. Since $\langle V, uu^\top \rangle = u^\top V u$, the latter condition is exactly the definition of positive semidefiniteness, which gives the assertion. ◀

A polynomial that is a sum of squares of polynomials from $\mathbb{R}[x]$ is called a *SOS polynomial*. We introduce the closed convex cone

$$\Sigma_B := \left\{ \sum_{i=1}^t f_i^2 : t \in \mathbb{N}, f_1, \dots, f_t \in \mathbb{R}[x]_B \right\}$$

of sparse SOS polynomials with the sparsity determined by B . Clearly, $\Sigma_B \subseteq \mathcal{P}_{B+B}(\mathbb{R}^n)$, but the inclusion is strict in general.

► **Proposition 48.** *Let $B \subseteq \mathbb{N}^n$ be a finite set of m exponent vectors. Then, Σ_B is the image*

$$\Sigma_B = \{(x^B)^\top M x^B : M \in \mathbb{S}_+^m\} \quad (21)$$

of \mathbb{S}_+^m under the linear map $M \in \mathbb{S}_+^m \mapsto (x^B)^\top M x^B \in \mathbb{R}[x]_{B+B}$, while Σ_B^* is given with $M_B = x^B (x^B)^\top$ by

$$\Sigma_B^* = \{v \in \mathbb{R}^{B+B} : L_v(M_B) \succeq 0\}, \quad (22)$$

which means that Σ_B^* is linearly isomorphic to a cross-section of Σ_+^m by a linear sub-space, with the isomorphism determined by the injective map $v \in \mathbb{R}^{B+B} \mapsto L_v(M_B) \in \mathbb{S}_+^m$.

Proof. A square of a polynomial $f(x) = f^\top x^B \in \mathbb{R}[x]_B$ can be written as

$$f(x)^2 = (f^\top x^B)((x^B)^\top f) = f^\top x^B (x^B)^\top f = f^\top M_B f = \langle M_B, f f^\top \rangle,$$

where in vector expressions involving f , we use the identification $\mathbb{R}[x]_B \leftrightarrow \mathbb{R}^B$. Consequently, $f_1^2 + \dots + f_t^2 = \langle M_B, Q \rangle$, where $Q = \sum_{i=1}^t f_i f_i^\top \succeq 0$. Conversely, every symmetric positive-semidefinite matrix Q of size $|B|$ can be decomposed into $\sum_{i=1}^t f_i f_i^\top$ for some f_1, \dots, f_t , which shows that $\langle M_B, Q \rangle = \sum_{i=1}^t f_i^2 \in \Sigma_B$. This shows (21). Since \mathbb{S}_+^m is self-dual, a symmetric matrix M is positive semidefinite if and only if $\langle M, Q \rangle \geq 0$ holds for every symmetric positive semidefinite matrix Q . We thus obtain

$$\begin{aligned} \{v \in \mathbb{R}^{B+B} : L_v(M_B) \succeq 0\} &= \{v \in \mathbb{R}^{B+B} : \langle L_v(M_B), Q \rangle \geq 0 \text{ for all } Q \succeq 0\} \\ &= \{v \in \mathbb{R}^{B+B} : L_v(\langle M_B, Q \rangle) \geq 0 \text{ for all } Q \succeq 0\} \\ &= \{v \in \mathbb{R}^{B+B} : L_v(g) \geq 0 \text{ for all } g \in \Sigma_B\} \\ &= \Sigma_B^*. \end{aligned}$$

which concludes the proof. ◀

In most cases, we do not have an explicit description in any of the available optimization paradigms for the cones $\mathcal{P}_{B+B}(\mathbb{R}^n)$ which we use in (D-P-RLX). In contrast, we have semidefinite representations for the cones Σ_B , as explained by Proposition 48. Thus, SOS cones along with the so-called positivstellensätze, which explain how positivity of polynomials can be certified in terms of SOS polynomials, provide a theoretical foundation for semidefinite approaches to polynomial optimization. Such approaches were pioneered by Lasserre [36].

For $f \in \mathbb{R}[x]_{B+B}$, we call $\sup\{f \in \mathbb{R}[x]_{B+B} : f - \lambda \in \Sigma_B\}$ an *SOS relaxation* of the unconstrained problem $\inf_{x \in \mathbb{R}^n} f(x)$ with respect to the exponent set B . Furthermore, we call $\sup\{L_v(f) : v_0 = 1, L_v(M_B) \succeq 0\}$ the *semidefinite moment relaxation* of $\inf_{x \in \mathbb{R}^n} f(x)$ with respect to the exponent set B . The dense versions of these relaxations are obtained by taking $B = \mathbb{N}_d^n$.

We formulate the duality between SOS and a semidefinite moment relaxation for the unconstrained polynomial optimization problem.

► **Proposition 49.** *For a finite exponent set $B \subseteq \mathbb{N}^n$ and $f \in \mathbb{R}[x]_{B+B}$, we have*

$$\sup\{\lambda \in \mathbb{R} : f - \lambda \in \Sigma_B\} = \inf\{L_v(f) : v_0 = 1, L_v(M_B) \succeq 0\}.$$

Proof. The assertion follows by applying Proposition 31 to the pointed closed convex cone $C = \Sigma_B$ and using the description (22) of $(\Sigma_B)^*$. ◀

From a computational perspective, the issue with the primal-dual pair of problems in Proposition 49 is that they involve semidefinite constraints with matrices of size $|B|$. When $|B|$ is large, the respective computations can become intractable, as discussed in Section 1.4. To cope with this, one can establish a sparse version of Proposition 49.

► **Proposition 50.** *Let $B_1, \dots, B_N \in \mathbb{N}^n$ be finite exponent sets and let $f \in \mathbb{R}[x]_A$ with $A = (B_1 + B_1) \cup \dots \cup (B_N + B_N)$. Then*

$$\sup\{\lambda \in \mathbb{R} : f - \lambda \in \Sigma_{B_1} + \dots + \Sigma_{B_N}\} = \inf\{L_v(f) : v_0 = 1, L_v(M_{B_i}) \succeq 0 \ \forall i \in [N]\}.$$

Proof. Clearly, $\Sigma_{B_1} + \dots + \Sigma_{B_N}$ is a subset of the pointed convex cone $\mathcal{P}_A(\mathbb{R}^n)$, so that the assumptions of Proposition 33 are true for $C_i = \Sigma_{B_i}$. Hence, the assertion follows from Proposition 33 by taking into account the expression (22) for the dual of the SOS cone. Note that we need to dualize Σ_{B_i} as a cone in $\mathbb{R}[x]_A$. In this case, (22) is only slightly modified to $(\Sigma_{B_i})^* = \{v \in \mathbb{R}^A : L_v(M_{B_i}) \succeq 0\}$, by using \mathbb{R}^A rather than $\mathbb{R}^{B_i+B_i}$. ◀

The primal-dual pair from Proposition 50 is a relaxed model for the pair of the problems (D-P-RLX) and (C-P-RLX) using the cones $\Sigma_{B_i} \subseteq \mathcal{P}_{B_i+B_i}(\mathbb{R}^n)$ and $\{v \in \mathbb{R}^{B_i+B_i} : L_v(M_{B_i}) \succeq 0\} \supseteq \mathcal{C}_{B_i+B_i}(\mathbb{R}^n)$. Thus, Proposition 50 corresponds to a pattern relaxation for the pattern family $\mathcal{F} = \{B_i + B_i : i \in [N]\}$.

We call the two problems in Proposition 50 the *sparse SOS relaxation* and the *sparse semidefinite moment relaxation* of $\inf_{x \in \mathbb{R}^n} f(x)$ with respect to the exponent sets B_1, \dots, B_N .

3.4.3 TSSOS

The intention of the TSSOS approach [81] is to sparsify SOS relaxations without losing quality. That is, one wants to replace an SOS relaxation by an equivalent sparse SOS relaxation. Assume we want to solve the unconstrained problem $\inf_{x \in \mathbb{R}^n} f(x)$ with $f \in \mathbb{R}[x]_A$. We fix an SOS relaxation with respect to some set B satisfying $A \subseteq B + B$. Having A and B , we now establish a sparse SOS relaxation with respect to some B_1, \dots, B_N that is equivalent to the SOS relaxation with respect to B for all polynomials f with the support contained in A . That is, one wants to have

$$\inf\{\lambda \in \mathbb{R} : f - \lambda \in \Sigma_B\} = \inf\{\lambda \in \mathbb{R} : f - \lambda \in \Sigma_{B_1} + \dots + \Sigma_{B_N}\} \ \forall f \in \mathbb{R}[x]_A. \quad (23)$$

Whenever the maximum size of the B_i 's is smaller than the size of B , the sparse SOS relaxation is usually more tractable than the SOS relaxation with respect to B .

The TSSOS approach is an iterative algorithm that gets A and B with $A \subseteq B + B$ as an input and performs updates on a partition $\mathcal{B} = \{B_1, \dots, B_N\}$ of B . In each iteration, the partition gets coarser (that is, N gets smaller), and the algorithm terminates in finitely many steps, providing a partition \mathcal{B} that satisfies (23).

To explain the algorithm, we use undirected graphs and we view a graph as a set of its edges. For example, $G = \{\{0, 1\}, \{1, 2\}, \{1, 3\}\}$ is a graph with three edges $\{0, 1\}$, $\{1, 2\}$ and $\{1, 3\}$ between nodes from the vertex set $\{0, 1, 2, 3\}$.

The algorithm starts by fixing $S_0 := A \cup 2B$. In the i -th iteration one considers the graph $G_i := \{\{\alpha, \beta\} : \alpha, \beta \in \mathbb{N}^n, \alpha + \beta \in S_{i-1}\}$. One defines

$$S_i = \{\alpha + \beta : \alpha, \beta \text{ nodes of } G_i \text{ connected by a path}\},$$

and sets the partition \mathcal{B}_i to be the family of connected components (vertex sets) of G_i . \mathcal{B}_i stabilizes after finitely many iterations and upon stabilization one has (23) with $\mathcal{B}_i = \{B_1, \dots, B_N\}$.

► **Remark 51.** How much one has gained by employing TSSOS, depends crucially on whether the size of the sets B_1, \dots, B_N is small. Understanding which choices of A lead to sparse TSSOS reformulations $\Sigma_{\mathcal{B}} \cap \mathbb{R}[x]_A$ is an interesting open problem.

► **Remark 52.** The basic TSSOS approach was generalized to constrained polynomial optimization and modified to the so-called Chordal-TSSOS approach [81] and the CS-TSSOS approach.

► **Remark 53.** A refined version of the TSSOS method has recently been suggested [66]. In rTSSOS, refined partitions are determined via integer programming, allowing a better fine-tuning between approximation quality and computational costs.

3.4.4 Optimization of sparse univariate polynomials over \mathbb{R}_+

Averkov and Scheiderer have shown that optimization of sparse univariate polynomials over \mathbb{R}_+ can be formulated exactly via a sparse SOS relaxation [9, Proposition 4.3].

► **Theorem 54.** For every univariate polynomial $f \in \mathbb{R}[x]_A$, where $A \subseteq \mathbb{N}$ is a finite set of exponents containing 0, of odd size $|A| = 2k + 1$, and with $d = \max(A) > 2k$ one has

$$\begin{aligned} \inf_{x \in \mathbb{R}_+} f(x) &= \sup\{\lambda : f - \lambda \in x^0 \Sigma_B + \cdots + x^{d-2k} \Sigma_B\} \\ &= \inf\{L_v(f) : v_0 = 1, L_v(x^i M_B) \succeq 0 \text{ for all } i \in 0, \dots, d - 2k\}, \end{aligned}$$

where $B = \{0, \dots, k\}$.

Theorem 54 implies that if a univariate polynomial f has $2k + 1$ terms and an arbitrarily large degree d , then the pattern relaxation with respect to the family of patterns $\{i, \dots, i + 2k\}$, where $0 \leq i \leq d - 2k$, for the optimization of f on the domain \mathbb{R}_+ gives the exact optimal value of (POP).

3.5 Lifted conic formulations

In Subsection 3.4, we discussed conic relaxations of polynomial optimization problems that are based on conic lifted representations. In this subsection, we present some abstract framework of lifted formulations and provide a summary of how the different liftings from Subsection 3.4 are related to each other.

3.5.1 Abstract lifted formulations

► **Definition 55.** Consider a closed convex cone $D \subseteq \mathbb{R}^n$. We say that a subset S of the finite-dimensional vector space \mathbb{R}^m over \mathbb{R} admits a D -lifted representation if

$$S = \{Mx : x \in D, Ax = b\} \tag{24}$$

for some linear maps $M \in \mathbb{R}^{m \times n}$, $A \in \mathbb{R}^{k \times m}$ and $b \in \mathbb{R}^k$. Geometrically, this means that S is a linear image of an affine slice of D .

► **Remark 56.** Representing S as an image $S = MD = \{Mx : x \in D\}$ of some cone D is a special case of a D -lifted formulation with $k = 0$. Furthermore, representing S as a cross-section $S = \{x \in D : Ax = b\}$ of D by the affine subspace $\{x : Ax = b\}$ is a special D -lifted formulation with M being the identity matrix.

Another way of writing the constraints $x \in D$, $Ax = b$ is by parametrizing the affine space $\{x : Ax = b\}$ as $x = v - Uy$ for some matrix U and a vector v . This allows representing the constraint $x \in D$, $Ax = b$ as $v - Uy \in D$ and Mx is represented as $Mx = Mv - MUy$.

When S in Definition 55 is a convex cone, one can replace b by the zero vector in the D -lifted representation of S so that S is represented as a linear image of a slice of D by a *linear* subspace.

In Subsection 3.4 we had a number of situations, in which we replaced the cones $\mathcal{C}_{P_i}(X)$ in (C-P-RLX) by smaller cones C_i that are images of some cones D_i under a linear map. The following abstract proposition underpins the duality for the respective sparse conic relaxation of (C-P-RLX).

► **Proposition 57.** Let $p, q \in \mathbb{R}^n$, let $C_1, \dots, C_N \subseteq \mathbb{R}^n$ be pointed closed convex cones and $C := C_1 + \cdots + C_N$, a pointed closed convex cone that does not contain q . Assume that for each $i \in [N]$ the cone C_i admits a formulation $C_i := M_i D_i := \{M_i y : y \in D_i\}$ as an image of a closed convex cone D_i under a matrix M_i . Then the problem $\sup\{\lambda \in \mathbb{R} : p - \lambda q \in C\}$ can be formulated as

$$\begin{aligned} \sup\{\lambda \in \mathbb{R} : p - \lambda q \in M_1 D_1 + \cdots + M_N D_N\} \\ = \inf\{\langle p, v \rangle : \langle q, v \rangle = 1, M_i^\top v \in D_i^* \ \forall i \in [N]\}. \end{aligned}$$

Proof. The assertion follows from Proposition 33 by taking into account that $(M_i D_i)^* = \{v \in \mathbb{R}^n : M_i^\top v \in D_i^*\}$. ◀

3.5.2 Examples of primal dual pairs with lifting

We look at some special cases of duality.

► **Proposition 58.** Let polynomials $g_1, \dots, g_s, f \in \mathbb{R}[x]_B$ be given for some finite set $B \subseteq \mathbb{N}^n$ with $0 \in B$. Then

$$\begin{aligned} \sup\{\lambda \in \mathbb{R} : f - \lambda \in g_1 \mathbb{R}_+ \cdots + g_s \mathbb{R}_+\} \\ = \inf\{L_v(f) : v \in \mathbb{R}^B, v_0 = 1, L_v(g_i) \geq 0 \ \forall i \in [s]\} \end{aligned}$$

and this primal-dual pair of problems provides a lower bound on the problem of the minimization of $f(x)$ subject to $g_1 \geq 0, \dots, g_s \geq 0$.

Proof. We use Proposition 57, which explains general sparse duality with lifting, for $A_i = g_i$ and $D_i = \mathbb{R}_+$, where we see $g_i(x)$ as a column vector in \mathbb{R}^B . The minimization problem involves the constraints $A_i^\top v \in D_i^*$, where $A_i^\top v = \langle g_i, v \rangle = L_v(g_i)$ and $D_i^* = (\mathbb{R}_+)^* = \mathbb{R}_+$. This gives the assertion. \blacktriangleleft

► **Remark 59.** The primal-dual pair in Proposition 58 is a pair of linear problems. If one knows some inequalities h_1, \dots, h_N valid on $X \subseteq \mathbb{R}^n$, one can use these linear problems to relax $\inf_{x \in X} f(x)$. In particular, it is possible to add the dual linear problem to the relaxations arising from the expression trees, see Section 2.2.2.

► **Remark 60.** Proposition 58 in combination with Handelman's theorem 10 gives a primal-dual pair of linear problems for the minimization $\inf_{x \in K} f(x)$ of a polynomial f over a polytope $K \subseteq \mathbb{R}^n$. If $g_1 \geq 0, \dots, g_m \geq 0$ is a system of linear inequalities that describes K , one can choose $h_i = g^{\alpha(i)}$ with $\alpha(i) \in \mathbb{N}^n$. Handelman's theorem guarantees that when $\{\alpha(i) : i \in [N]\} = \mathbb{N}_d^n$ and d is large enough, the primal-dual pair approximates $\inf_{x \in K} f(x)$ arbitrarily well.

► **Remark 61.** For the primal and dual SAGE relaxation, based on a family of SAGE patterns P_1, \dots, P_N with $B := P_1 \cup \dots \cup P_N$, the cones D_i are GMC cones $\text{GMC}_{\lambda(i)}$ with $\lambda(i) \in \mathbb{R}_{>0}^{k_i+2}$, and the matrix A_i provides a coordinate embedding of $\mathbb{R}^{k_i+2} \cong \mathbb{R}^{P_i}$ into \mathbb{R}^B that bijectively sends \mathbb{R}^{P_i} to $\mathbb{R}^{P_i} \times \{0\}^{B \setminus P_i}$ by appending zero components. Accordingly, A_i^\top describes the coordinate projection of \mathbb{R}^B onto $\mathbb{R}^{P_i} \cong \mathbb{R}^{k_i+2}$. Analogous comments also apply to the SONC, SDSOS, and sparse SOS relaxations. The underlying cone for the SDSOS relaxation is $\text{GMC}_{1/2,1/2}$. For sparse SOS relaxations, Proposition 57 provides an abstract version of Proposition 50, where A_i corresponds to the linear map $Q \in \mathbb{S}^{m_i} \mapsto (x^{B_i})^\top Q x^{B_i} \in \mathbb{R}[x]_A$, with $m_i = |B_i|$, and A_i^\top corresponds to the adjoint of this map, which is the map $v \in \mathbb{R}^A \rightarrow L_v(M_{B_i}) \in \mathbb{S}^{m_i}$.

3.5.3 Relevant conic-optimization paradigms

► **Remark 62.** Conic programming with respect to the cone \mathbb{R}_+^n is linear programming. A \mathbb{R}_+^n -lifted formulation is also called a *linear extended formulation*. It is known that a set X admits a linear extended formulation if and only if X is a polyhedron. This follows from the fact that polyhedrality is preserved under taking linear images. Thus, linear programming convexification techniques correspond to conic optimization for the cone \mathbb{R}_+^n .

► **Definition 63.** We call $(\mathbb{S}_+^2)^m$ a second-order cone and we call a set S second-order cone representable if it admits a $(\mathbb{S}_+^2)^m$ -lift. We define second-order cone programming as optimization of a linear objective function subject to finitely many LMIs $M_1(x) \succeq 0, \dots, M_m(x) \succeq 0$ with matrices of size 2.

► **Remark 64.** Polyhedra are second-order cone representable, since every linear inequality $\langle a, x \rangle + b \geq 0$ in the variables x can be rewritten as a linear matrix inequality $\begin{pmatrix} \langle a, x \rangle + b & 0 \\ 0 & 0 \end{pmatrix} \succeq 0$ with a matrix of size 2. On the other hand, not every second-order cone representable set is a polyhedron: consider for example the unit disc, described as $\begin{pmatrix} 1+x_1 & x_2 \\ x_2 & 1-x_2 \end{pmatrix} \succeq 0$.

► **Definition 65.** A \mathbb{S}_+^m -lifted formulation is also called a semidefinite lifted formulation or a semidefinite extended formulation. We call a set that admits a semidefinite extended formulation semidefinitely representable.

► **Remark 66.** Every second-order cone representable set is semidefinitely representable, because $(\mathbb{S}_+^2)^m$ corresponds to a cross-section of \mathbb{S}_+^{2m} . On the other hand, Hamza Fawzi showed that there exist semidefinitely representable sets that are not second-order cone representable [22]. This result was generalized by Averkov [8] and later generalized even further by Saunderson [62].

There is a well-developed theory of computational methods for solving linear, second-order cone and semidefinite optimization problems, with the methods implemented by various solvers. So, for each relaxation of a polynomial optimization problem that relies on linear, second-order cone or semidefinite lifted formulations, it is possible to use the available computational machinery to carry out computations in practice.

► **Remark 67.** For SAGE and SONC relaxations we use lifts with respect to the Cartesian products

$$\text{GMC}_{\gamma(1)} \times \dots \times \text{GMC}_{\gamma(m)}$$

of the GMC cones. For conic programming with respect to GMC cones computational approaches are also available. Alternatively, one can pass from GMC cones to second-order cones, using a result of Ben-Tal and Nemirovski [12], which shows that GMC_λ is second-order cone representable.

4 Conclusions and Outlook

We have presented different aspects of convexification and sparsity exploitation for primal and dual approaches to polynomial optimization in a self-contained review. The review contains novel proofs. For example, we proved the dual positivstellensätze from Putinar and Handelman using infinite-dimensional matrices together with a projection argument, which allowed omitting the introduction of moments. Our generalized framework is meant to facilitate the transfer of theoretical results to special cases.

The formal convexification introduced in Section 1.2 is a step which is not always explicitly articulated when a particular approach is pursued. Instead, authors might go directly into details and blend this step with their favorite version of convex optimization. In contrast, the simple and general idea of Section 1.2 allows us to have duality explained for different kinds of methods, affecting and simplifying other parts of the paper. E.g., we included also LP-based relaxations relying on Handelman's theorem and convexification from expression trees in our survey. While Handelman's theorem is well known to experts in real algebra, the fact that its dual version plays a role in the global NLP community is something worth emphasizing. We presented moment vs. SOS duality and the Handelman vs. bound-factor duality in one source to raise mutual awareness.

Our numerical experiments in the Appendix indicate the huge potential of tailored relaxations based on pattern families. Structure exploitation may bring huge advantages in comparison to standard approaches both in terms of approximation quality and in computational runtime. Patterns have already been used for special cases. Ahmadi and coworkers [2] investigated a combination of the patterns \mathbb{N}_2^n and axis-parallel chains for the unconstrained case $X = \mathbb{R}^n$. They were able to show equivalences of primal and dual formulation and exact representations for specific exponent sets A . Another contribution at the interface between robotics and polynomial optimization [32] also exploits particular types of patterns. It uses term and correlative sparsity and adjusts TSSOS to improve performance.

Ideally, one would like to develop an adaptive method that works best for any type of instances, i.e., with respect to exponents $\alpha \in A$ and coefficient vectors. Detecting favorable structures a priori in an automatized, adaptive way, seems to be one of the future challenges in polynomial optimization. First ideas in this direction are based on graph algorithms [56] or integer programming [66], but require further research. An integration into concepts from the global NLP community, in particular branch-and-bound frameworks, is also very promising.

Appendix

A Notation

We denote the convex hull and the convex conic hull of sets by conv and cone , respectively. The transposition operation is denoted as \top . By default, vectors are interpreted as columns, but when the choice of the column vs. row convention does not matter, we write vectors as rows to avoid unnecessary transposition.

$[n]$	$\{1, \dots, n\}$
$ A $	Cardinality of set A
A	Finite exponent set, $A \subseteq \mathbb{N}^n$
B	Finite exponent set in \mathbb{N}^n , $B \supseteq A$ with $B = P_1 \cup \dots \cup P_N$
$A + B$	Minkowski sum, $\{\alpha + \beta : \alpha \in A, \beta \in B\}$
$\text{BF}(B)$	Polyhedral relaxation based on (14)
$\text{Box}(l, u)$	Box of lower and upper bounds $[l_1, u_1] \times \dots \times [l_n, u_n]$
$\mathcal{C}_A(X)$	Moment cone $\overline{\text{cone}\{x^A : x \in X\}}$
$\text{CRLX}_{\mathcal{F}}(X)$	Conic moment relaxation $\{v \in \mathbb{R}^B : v_{P_i} \in \mathcal{C}_{P_i}(X) \text{ for all } i \in [N]\}$
$\text{deg}(A)$	Degree, $\max\{\text{deg}(x^\alpha) : \alpha \in A\}$
$\mathcal{M}_A(X)$	Moment body $\mathcal{M}_A(X) := \text{conv}\{(x^\alpha)_{\alpha \in A} : x \in X\}$
\mathcal{F}	Pattern family $\mathcal{F} = \{P_1, \dots, P_N\}$
$f(x)$	Objective function, polynomial
g	Vector of polynomial inequalities $g(x) \geq 0$ defining X
h	Vector of polynomial equalities $h(x) = 0$ defining X
K	Special case of feasible set X , either box or polytope
$L_v(f)$	Linearization map $L_v\left(\sum_{\alpha \in A} f_\alpha x^\alpha\right) := \sum_{\alpha \in A} f_\alpha v_\alpha$
$M_B(x)$	$M_B(x) = x^B (x^B)^\top$
$\text{MR}(\mathcal{B}_0, \dots, \mathcal{B}_k)$	$\{v \in \mathbb{R}^P : L_v(g_i M_B) \succeq 0 \ \forall i \in [k] \text{ and } B \in \mathcal{B}_i\}$
\mathbb{N}_d^n	Space of exponents of maximum degree d for polynomials in \mathbb{R}^n
P_i	Pattern, as a member of the pattern family \mathcal{F}
$\mathbb{R}[x]$	Ring of polynomials in variables $x \in \mathbb{R}^n$ with real coefficients
$\mathbb{R}[x]_A$	As $\mathbb{R}[x]$, but with monomials whose exponent vectors are in A
\mathbb{R}^A	Real vector space isomorphic to $\mathbb{R}^{ A }$ with vectors indexed by A
\mathbb{S}^m	Space of symmetric matrices of size m over reals
\mathbb{S}_+^m	Space of positive semidefinite matrices in \mathbb{S}^m
s	Number of polynomial inequality constraints in X
t	Number of polynomial equality constraints in X
v	Monomial variable vector with entries $v_\alpha = x^\alpha \in \mathbb{R}$
x	Variables of original problem (POP), $x \in \mathbb{R}^n$
x^α	Product $x^\alpha = \prod_{i=1}^n x_i^{\alpha_i}$
X	Feasible set, $x \in X \subseteq \mathbb{R}^n$
\bar{Y}	Topological closure of set Y
y_A	If $y \in \mathbb{R}^n$ and $B \subseteq \mathbb{N}^n$, then for $A \subseteq B$ the vector y_A is the projection of y on coordinates indexed by A : $y_A = (y_\alpha)_{\alpha \in A} \in \mathbb{R}^A$
Y_A	Projection of a set Y on the A coordinates: $Y_A = \{y_A : y \in Y\}$

B Software tools that support sparse SDP based POP

We briefly survey examples of polynomial optimization tools that are based on semidefinite optimization and positivstellensätze to orient the reader towards the possibilities of exploiting sparsity in concrete applications.

The landscape of SDP is generally divided into two categories: modeling languages that allow you to write optimization problems in a human-readable way, and solvers that are the numerical workhorses that actually compute the solution.

The solvers comprise general purpose interior point solvers that are robust, high-precision solvers suitable for small to medium-sized problems with thousands of variables. Popular examples are MOSEK [50], SeDuMi [69], and SDPT3 [75]. More specialized solvers use first-order methods or decomposition techniques to handle problems that are too large for standard interior-point methods. Examples are SCS [54], Clarabel [26], or COSMO [23].

The modeling languages or user interfaces comprise YALMIP [42], a versatile modeling language for MATLAB, supporting almost all available SDP solvers; CVX [27] and CVXPY [17], a disciplined convex programming approach; Gloptipoly 3 [30], a MATLAB-based tool that uses the Lasserre hierarchy of moment relaxations for global optimization of multivariate polynomials; SparsePOP [80], a tool specifically designed to exploit structured sparsity using chordal graph theory; and JuMP [43], the standard modeling language for the Julia ecosystem and `sumofsquares.jl` [40], the primary library in the Julia ecosystem for SOS programming that leverages the JuMP framework and supports a wide variety of solvers like MOSEK, SCS, and Clarabel. The exploitation of term sparsity is emphasized in the Julia package TSSOS [44]. Also in PYTHON modeling tools are available, e.g., `SumOfSquares.py` [83] and SPOT [33].

Providing sparsity information in the spirit of monomial patterns is possible in most of these tools, but depends on the particular modeling language.

Gloptipoly 3 [30] is very flexible in setting up constraints. It can generate moment matrices to be used in SDP constraints, which have a user-defined layout, and it can work with moments of not just one measure, but of several measures as specified in Section 5 of the documentation. So, Gloptipoly 3 supports sparse constraints in the spirit of our discussion natively, and it approaches these constraints from the moment side.

YALMIP [42] is a Matlab based modeling language for different sorts of optimization problems. It provides the function `solvemoment` to set up and solve moment relaxations. So, its preferred side of duality is the moment side. According to the documentation¹ YALMIP does not seem to provide an intrinsic support for sparse semidefinite relaxations. YALMIP supports many other optimization paradigms, so moment relaxations is just one of many aspects of this versatile package.

SparsePOP [80] is implemented in Matlab and C++. According to documentation² this package seems to handle SDP relaxations from the SOS perspective and focus on correlative sparsity, i.e., the sparsity depending on the graph of the set of all variables x_1, \dots, x_n , with two variables connected by an edge if they both occur simultaneously in a monomial.

The software `sumofsquares.jl` [40] favors the dual SOS approach and is flexible in setting up constraints. It can introduce sparse polynomials, for which the SOS-condition can be added as the constraint to the problem. Thus, this package too supports flexible sparsity in the spirit of our paper³. The choice of the side of the duality does not seem that relevant. It just reflects the way the developers prefer to think about semidefinite relaxations of POP. Eventually, SDP solvers solve the primal-dual pair so that it does not matter what side of the pair is fed into the SDP solver. We also would like to mention that `sumofsquares.jl` allows one to work with other bases than the polynomial basis (which might be important to achieve stability in numerical computations). The package also directly supports the TSSOS and Chordal-TSSOS kinds of sparsity⁴.

We provide a simple example of how a monomial pattern decomposition can be used in `sumofsquares.jl` to prescribe a desired form of sparsity of SOS polynomials. For illustration, we want to minimize the polynomial

$$f = (1 + 2x - x^2 + x^3)^2 + 2(1 + 3y - 7y^2)^2 + x^2(2 - 3xy)^2 + 7(1 - x^3 + y)^2$$

in two variables x and y . We do this by defining four patterns A , B , C , and D , maximizing the value of λ and by enforcing the sum of pattern relaxations to be $f - \lambda$. This is the dual form of (P-RLX) as formulated as the bound-certification problem (4).

¹ <https://yalmip.github.io/command/solvemoment/>

² <https://sparsepop.sourceforge.io/>

³ <https://jump.dev/SumOfSquares.jl/stable/variables/#Polynomial-variables>

⁴ https://jump.dev/SumOfSquares.jl/stable/generated/Sparsity/term_sparsity/

```

1  using SumOfSquares
2  using DynamicPolynomials
3  using CSDP
4
5  # Set up model
6  solver = optimizer_with_attributes(CSDP.Optimizer, MOI.Silent() => false)
7  model = Model(solver)
8
9  @polyvar x y
10
11 # Four patterns to be used
12 A=monomials(x,0:3)
13 B=monomials(y,0:2)
14 C=[x,x^2*y]
15 D=[1, x^3, y]
16
17 # Define variables
18 @variable(model,a,SOSPoly(A))
19 @variable(model,b,SOSPoly(B))
20 @variable(model,c,SOSPoly(C))
21 @variable(model,d,SOSPoly(D))
22 @variable(model,lambda)
23
24 # Formulate constraint and objective
25 f=(1+2*x-x^2+x^3)^2 + 2*(1+3*y-7*y^2)^2 + x^2 * (2- 3*x*y)^2 + 7*(1-x^3+y)^2
26 @constraint(model,f==lambda + a + b + c + d)
27 @objective(model,Max,lambda)
28
29 # Optimize and output
30 optimize!(model)
31 objective_value(model)

```

The solver returns the solution 4.4436 after 17 iterations in the fraction of a second.

C Computations

In this section we shall investigate numerically the impact of various forms of convexification and sparsification on computational runtime and on the quality of approximations. We describe which algorithms were evaluated with which criteria in Section C.1, and describe the benchmark test set of polynomial optimization problems in Section C.2. The numerical results are then presented and discussed in Section C.3.

Real-world instances of POP that have been described in the literature are usually very specific in their structure and hence favor particular relaxation patterns. Understanding which patterns are particularly strong for applications such as optimal power flow, stability of dynamic systems, or max-cut is certainly interesting. The necessary in-depth-analysis has already been performed in other papers, resulting in particularly successful convexification strategies. See [45] for practical applications of POP. Our goal is not to suggest specific custom sparsity patterns that might outperform existing methods, but rather give a flavor of how the theoretical results of the paper carry over to numerical practice.

C.1 Convexification and sparsification approaches

We are interested in different ways of convexification and in a comparison to existing solvers for polynomial optimization problems based on either primal or dual approaches. In all of our numerical comparisons we used the following software. The abbreviations will be used in all following figures.

Method	Description
B	We used the software BARON 1.8.9 [61, 72] with default settings, called from MATLAB . BARON is a mature solver, based on branch-and-bound and polyhedral relaxations. We enforced a 1000 second time limit, as discussed further below.
R	Root node relaxation of BARON , indicating the computational costs and approximation quality of the polyhedral relaxation without branch-and-bound based refinements.
Y	We used the software YALMIP20200930 [42] as a MATLAB toolbox. It allows computing the moment as well as the SOS relaxation of (POP). We used YALMIP 's solvesos at the lowest possible level of the SOS hierarchy with MOSEK9.2.32 [50].
SOS	In study C.3.3, method Y could not provide a solution due to memory issues. We replaced it with a custom implementation of Lasserre's SDP relaxation at the lowest possible level of the hierarchy. It gave qualitatively similar results on other test problems (data not shown).
CS	We used the software CS-TSSOS1.00 , based on the (dual) TSSOS approach [81], compare Section 3.4.3. CS-TSSOS is a JULIA package that allows exploiting correlative sparsity and term sparsity simultaneously. We called the first level of the hierarchy by running the command <code>cs_tssos_first</code> with settings <code>order = $\lceil \frac{\deg(f)}{2} \rceil$</code> and <code>TS="MD"</code> .
SIG	The PYTHON package SAGEOPT0.5.3 [15, 34] allows computing relaxations based on SONC polynomials as well as SAGE relaxations based on signomials. For SONC relaxation no finite lower bounds could be obtained for any of the instances, all shown results correspond to SAGE relaxations, compare Section 3.4.1. We used the method <code>sig_constrained_relaxation</code> on the lowest possible hierarchy level.

The rationale behind the selection of these methods was the variety of primal and dual approaches: **BARON** for global NLP approaches relying on polyhedral relaxations, **YALMIP** as an implementation of the dense Lasserre's semidefinite relaxation, **CS-TSSOS** as a particular way to sparsify the latter, and **SAGEOPT** as an alternative approach relying on the circuit patterns. Note that more solvers to solve (POP) globally exist, such as **SCIP** [78], **COUENNE** [11], or **LINDOGlobal**. As they are based on polyhedral relaxations, too, we concentrated on a comparison with **BARON**. We are not aware of any other source, where such a broad comparison across a wide range of approaches was carried out.

In different numerical studies in Section C.3 we compared the approaches listed above to implementations of pattern relaxations from Section 2 as specified below. The code for solving the pattern relaxation (P-RLX) was implemented and run in **MATLAB** 9.10.0.1669831 (R2021a) Update 5 [74], consists of roughly 3500 lines of code, and uses **MOSEK9.2.32** [50] to solve the relaxations. We used the following pattern relaxations in order of appearance.

Method	Description
M	Multilinear pattern relaxation using $\mathcal{F}_A^m := \{\{0, \alpha_1\} \times \cdots \times \{0, \alpha_n\} : \alpha \in A\}$. Here, and in what follows, as a preprocessing step we only keep the inclusion-maximal patterns of the family and leave out all the other patterns.
C	Chain relaxation for the pattern family $\mathcal{F}_A^c := \{C_\alpha \alpha : \alpha \in A\}$ given by $C_\alpha := \left\{ k \frac{\alpha}{\gcd(\alpha)} : k = 0, \dots, 2\lceil \gcd(\alpha)/2 \rceil \right\},$ <p>where $\gcd(\alpha)$ denotes the greatest common divisor of the components of α.</p>
H	Pattern relaxation for the pattern family \mathcal{F}_A^h constructed as follows. Choose the minimal $d \in \mathbb{N}$ such that $A \subseteq \{0, \dots, 2d\}^n$ and add to \mathcal{F}_A^h the patterns $\{k e_i : k = 0, \dots, 2d\}$ for each $i \in [n]$, $\{(k, \dots, k) : k = 0, \dots, 2d\}$, $\{0, k\} \times \cdots \times \{0, k\}$ for every $k = 0, \dots, 2d$, and all patterns from \mathcal{F}_A^m .
S	Pattern relaxation for the family $\mathcal{F}_A^s := \{S_{\alpha,i} : i \in [n], \alpha \in A\}$ $S_{\alpha,i} := \{(\alpha_1, \dots, \alpha_{i-1}, k, \alpha_{i+1}, \dots, \alpha_n) : k = 0, \dots, 2\lceil \alpha_i/2 \rceil\}$ <p>of shifted axis-parallel chains containing elements of A.</p>
MC	Pattern relaxation for the pattern family $\mathcal{F}_A^{mc} := \mathcal{F}_A^m \cup \mathcal{F}_A^c$.
T	Pattern relaxation for the pattern family \mathcal{F}_A^t consisting of truncated submonoid patterns constructed as follows. Fix a truncated submonoid pattern $P = 2\mathbb{N}_{2\lceil \deg(A)/4 \rceil}^n$. For each $\alpha \in \mathbb{N}^n$, consider the pattern $P_\alpha = \Gamma \mathbb{N}_{2\lceil \deg(A)/2 \rceil}^n$ <p>given by the matrix $\Gamma = (\gamma(1), \dots, \gamma(n))$ with $\gamma(i) = e_i$ if $\alpha_i > 0$ and $\gamma(i) = 0$ otherwise. Define $\mathcal{F}_A^t := \{P\} \cup \{P_\alpha : \alpha \in A \setminus P\}$. This pattern is designed to give more tractable Lasserre-type relaxations for low degrees d.</p>

C.2 Test set and evaluation criteria

Establishing a meaningful test set for evaluating approaches to (POP) is a nontrivial task. Regarding the choice of the objective f , there are low-degree and high-degree instances, there are dense instances, instances with a specific structural sparsity and instances with a random sparsity, instances with many variables and instance with a small number of variables, structured instances occurring in particular applications and randomly generated instances. There are many ways to combine the above properties so that a thorough evaluation would require a large number of different types of instances. Evaluation on such a diverse test set would allow us to understand on which types of instances a particular approach should be applied.

We are not aware of any library of polynomial optimization instances that exhibits such a desired degree of diversity. Thus, we randomly generated a test set of instances of different types and complemented it with some structured examples. Our test sets evaluates 12 choices of A , classified into four types: dense exponent sets, sparse sets, sparse sets with large dimension n and small degree d , and specifically structured sets. The latter is composed of exponent sets that are designed to give specific insight for some of the tested relaxation approaches.

In all calculations, we fix the feasible set X to the box $K = [0, 1]^n$. For each A , we create a sample of twenty polynomials by sampling coefficients for f using a uniform distribution in $[-1, 1]^A$. However, samples are only included in the test set, when they are sufficiently difficult. To quantify difficulty, we used the performance of method B. Only when either the minimization problem $\min_{x \in K} f(x)$ or the maximization problem $\min_{x \in K} -f(x)$ can not be solved completely by BARON within the time limit of 1000 seconds, do we keep f . This procedure explains why the reported runtimes of BARON are always at least 1000 seconds and only allow to indicate the potential of improvement of BARON for difficult cases. This preselection is motivated by the basic idea that we primarily need to improve the performance of methods on instances that are currently hard for established solution techniques.

The first evaluation criterium is the approximation quality of the relaxation. To consider different directions of optimization over the feasible set K , we consider minimization and maximization problem. We use the normalized value

$$\text{triv}_{\mathcal{F}}(f, K) = \frac{\max_{v \in \text{RLX}_{\mathcal{F}}(K)} L_v(f) - \min_{v \in \text{RLX}_{\mathcal{F}}(K)} L_v(f)}{\max_{v \in \text{Box}(l_A, u_A)} L_v(f) - \min_{v \in \text{Box}(l_A, u_A)} L_v(f)} \in [0, 1] \quad (25)$$

as an indicator of the tightness of the calculated relaxations in comparison to the trivial bounds obtained from the bounds (l_A, u_A) on the monomials occurring in f . Values close to 1 indicate that the bounds obtained via

relaxation $\text{RLX}_{\mathcal{F}}(K)$ do not improve much over the trivial bounds in the denominator, whereas values close to 0 indicate that $\text{RLX}_{\mathcal{F}}(K)$ is a good relaxation.

The second reported criteria is computational time, provided as the wall-clock time in seconds, averaged over all 20 samples. The solvers were run on a compute server with 4 Intel(R) Xeon(R) Gold 6138 CPUs with 20 cores of 2 threads and 1 TB RAM each under Ubuntu 20.04.3. Each solver-instance pair was assigned to one such job, i.e., the solvers themselves did not use the parallel structure. In order to distribute the solver-instance pairs to the 80 cores we used [70].

C.3 Numerical results

We investigate the performance of different methods with respect to approximation quality evaluated via (25) and computational run time. We structure the analysis according to properties of the exponent sets A .

C.3.1 Dense Exponent Sets

We first consider dense exponent sets $A = \mathbb{N}_d^n$ for $n \in \{2, 4\}$ and $d = 10$. Results are shown in Figure 3.

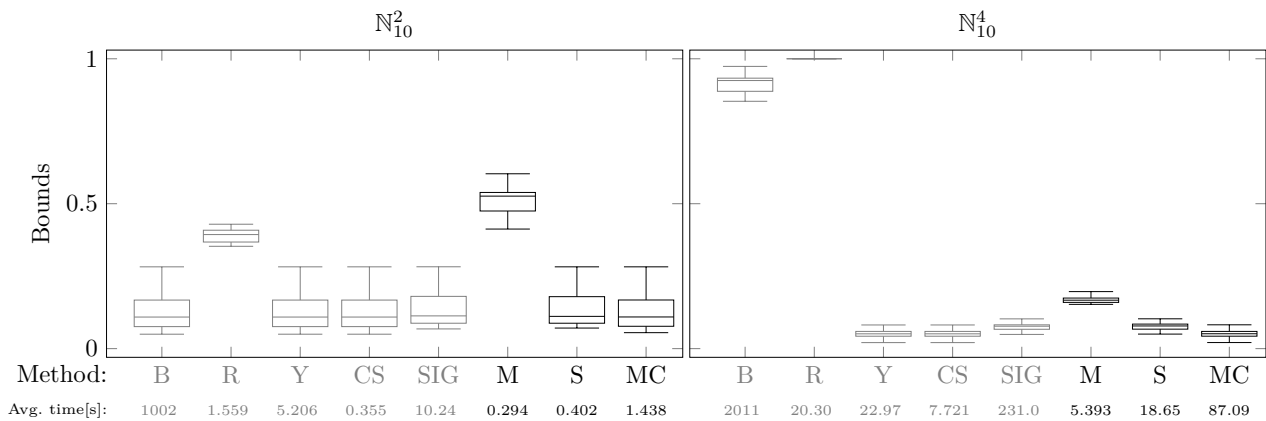


Figure 3 Results for dense exponent sets $A = \mathbb{N}_d^n$. Shown is the median and standard variation for 20 sample coefficient vectors f of the evaluation criterium (25) for the methods specified in Section C.1. Note that a value close to 0 indicates a good relaxation. The average computational time in seconds is provided below each method in the bottom row. Note that only instances are selected for which BARON needs the full time limit on either minimization or maximization problem.

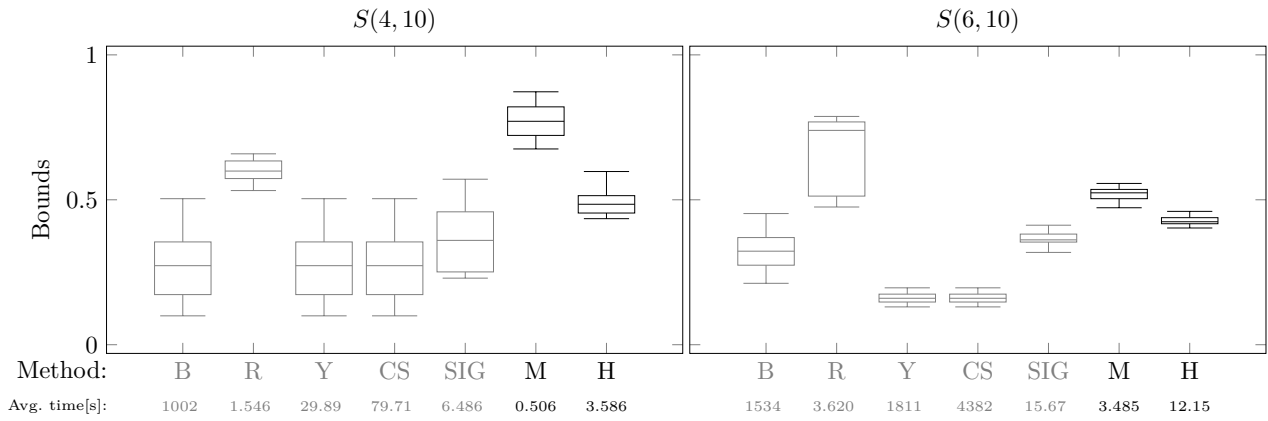
The root relaxation R of BARON can be solved in 1.5 and 20 seconds, respectively. The solution B using branch-and-bound is constrained on purpose by the imposed time limit of 1000 seconds for minimization and maximization problem. During this time a certain number of nodes can be processed, which leads to a significant reduction in the case \mathbb{N}_{10}^2 . For the higher dimensional polynomials in \mathbb{N}_{10}^4 the solution time per node is increased, though. The reduced number of nodes that can be processed in the time limit do only result in a modest improvement of the bounds.

The methods Y, CS, SIG, S, and MC provide good bounds. In comparison, the runtime for SIG is not very competitive. The multilinear polyhedral relaxation M can be solved fast, but results in weaker bounds, as expected. The bounds are better for $n = 4$ compared to R, though, probably due to a larger pattern size and hence more connections between monomial variables compared to the relaxation BARON uses.

C.3.2 Sparse Exponent Sets

We use randomly generated sparse exponent sets $A = S(n, d)$ to test pattern families that do not assume any structure of A . $S(n, d)$ is generated by randomly picking $\left\lceil \sqrt{\binom{n+d}{d}} \right\rceil$ exponents via `randperm` from \mathbb{N}_d^n . Results are shown in Figure 4.

In comparison to BARON, the semidefinite relaxations Y and CS provide very good results also in $n = 6$ dimensions, albeit at a high computational price. In comparison, SIG, M, and H provide weaker bounds, but up to three orders of magnitude faster. Method M based on multilinear patterns results in weak bounds, which can

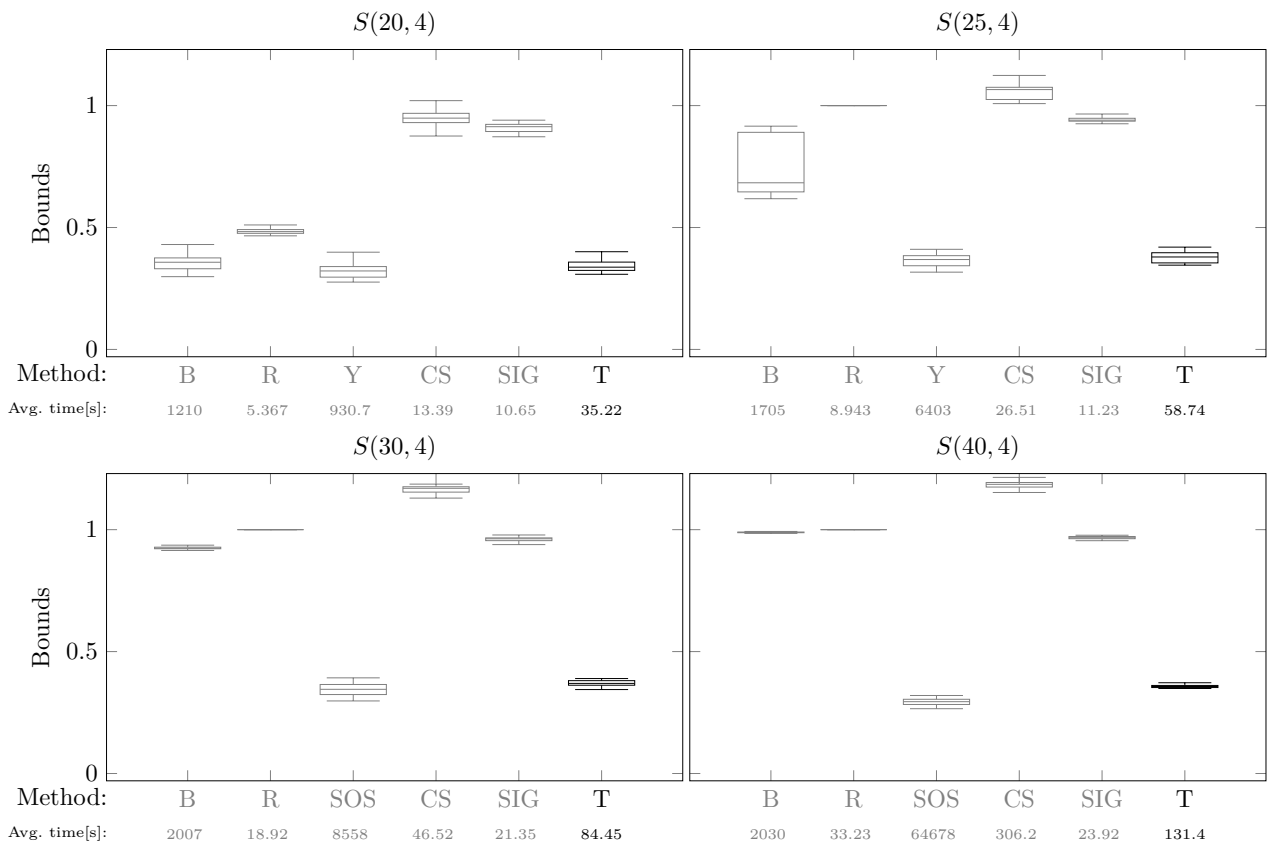


■ **Figure 4** As in Figure 3, but for sparse exponent sets $S(n, d)$.

be strengthened considerably with method H at some computational expense by additionally enforcing indirect connections between moment variables via $n + 1$ chains and d multilinear patterns.

C.3.3 Sparse Exponent Sets with $n > d$

As in Section C.3.2 we use sparse exponent sets, however now with a high number of variables $n = 20, 25, 30, 40$ and low degree $d = 4$. Results are shown in Figure 5.



■ **Figure 5** As in Figure 3, but for sparse exponent sets $S(n, d)$ with $n > d$.

The difficulty of these instances becomes apparent from the weak bounds close to 1 provide by method R and for $n = 40$ also for B. Remember that a value of 1 in (25) corresponds to relaxations that have no advantage over the trivial lower bounds on the monomial variables. Excellent bounds are provided by the Y / SOS methods. However, the computational time increases drastically and memory issues arise for Y, validating the theoretical

considerations in Section 1.4. It becomes clear that alternatives are necessary for problems in higher dimensions. It is interesting that methods CS and SIG run comparatively fast, but do not result in good bounds.

One such candidate is method T, which is tailored for problems with low degree and many variables and profits computationally from the small size of the involved LMIs. The very good bounds provided by T can be calculated two orders of magnitude faster than with YALMIP. This allowed computing nontrivial bounds for instances with exponent sets $S(80, 4)$ in approximately 400 seconds.

C.3.4 Specifically structured test sets

To complement the results on dense and sparse exponent sets, we consider specific exponent sets with particular properties. We start with adversarial instances. Assume for example that we want to solve (POP) with $A = \{(k, \dots, k) : k = 0, \dots, 2d\}$, where $d \in \mathbb{N}$. This is actually a disguised one-dimensional problem, since we can make a substitution $y = x_1 \dots x_n$. Thus, we need patterns that would link the monomial variables with the exponents in A . Since A is a chain, choosing A as a pattern is optimal. This means, C is an optimal method for this A . But if we use M, we would link the monomials $x_1^i \dots x_n^i$ with the variables x_1, \dots, x_n by convex constraints, which is not helpful. So the above support is an adversarial choice for method M. By looking at such kinds of special A 's we are able to understand which are the preferred choices of A for the methods we evaluate and on which choices of A the methods have difficulties. We test four exponent sets based on chain pattern families,

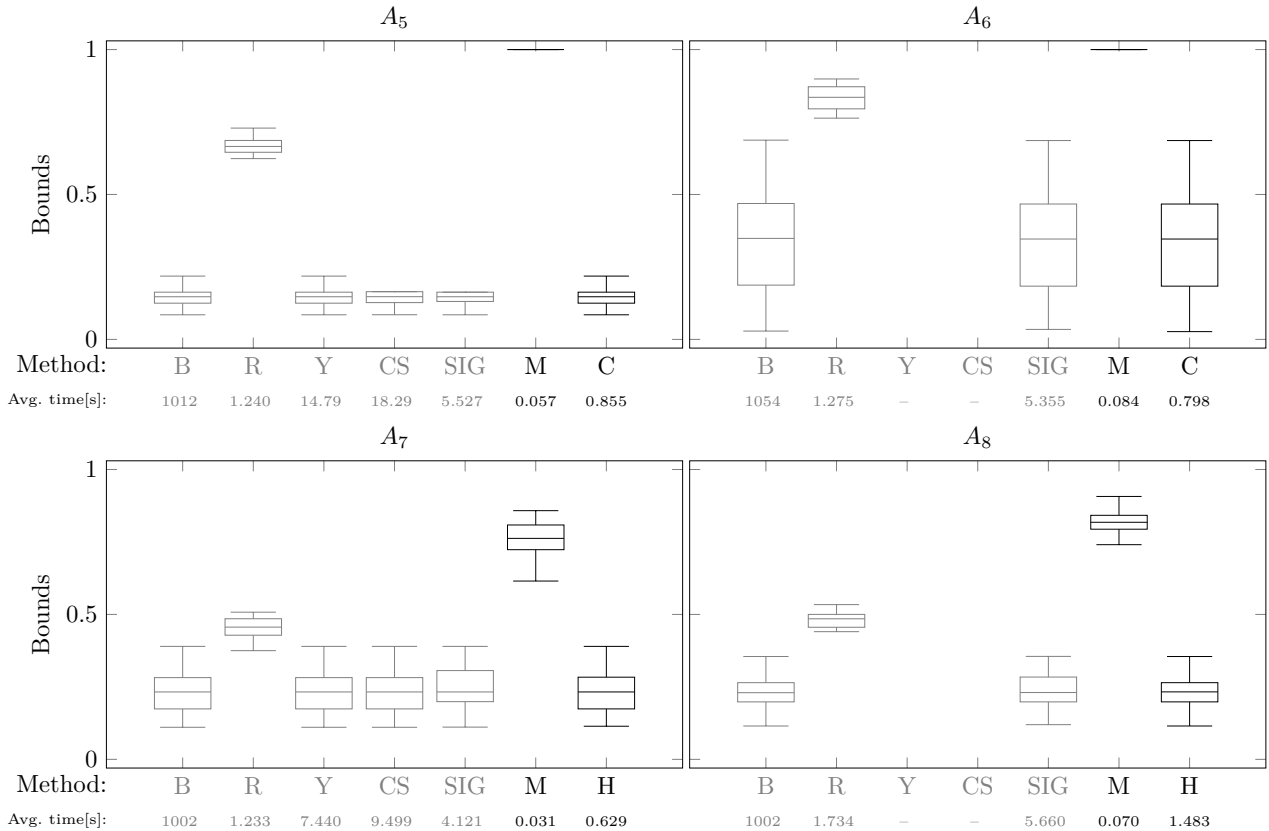
$$A_5 = \{(k, k) : k = 0, \dots, 10\} \subset \mathbb{N}_{10}^2$$

$$A_6 = \{(k, k, k, k) : k = 0, \dots, 10\} \subset \mathbb{N}_{10}^4$$

$$A_7 = \{k\alpha : k = 0, \dots, 10, \alpha \in \{e_1, e_2, e_1 + e_2\}\} \subset \mathbb{N}_{10}^2$$

$$A_8 = \{k\alpha : k = 0, \dots, 10, \alpha \in \{e_1, e_2, e_3, e_4, e_1 + e_2 + e_3 + e_4\}\} \subset \mathbb{N}_{10}^4$$

Results are shown in Figure 6.

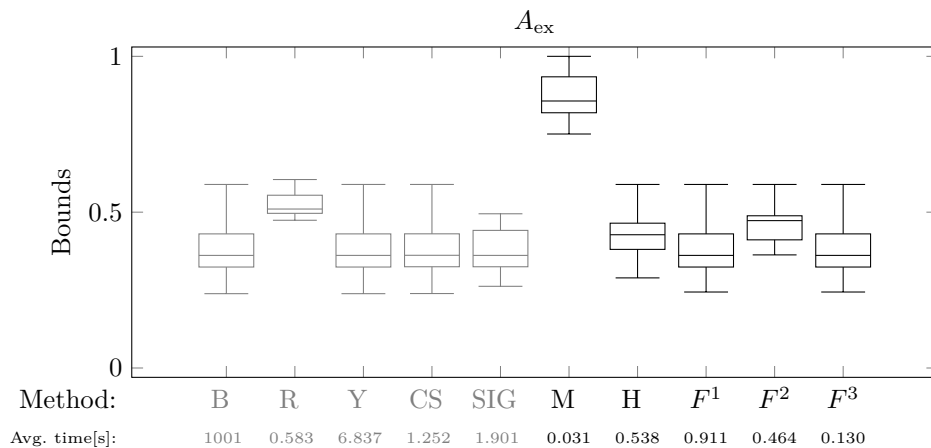


■ **Figure 6** As in Figure 3, but for specific adversarial exponent sets A_5, A_6, A_7, A_8 .

As a main take-away, the exploitation of chain patterns in C and H brings a huge advantage in comparison to the multilinear pattern method M. The resulting bounds are similar to those produced by B and SIG, but

significantly faster. This shows that the structure of the exponent set may be decisive for the performance of particular methods. Detecting such structures a priori in an automatized, adaptive way, seems to be one of the future challenges in polynomial optimization.

Findally, we study the exponent set A_{ex} as defined in Example 26 and visualized in Figure 2. In addition to the solvers and methods M and H, we also apply three custom pattern families F^1 , F^2 , and F^3 as introduced in the fifth row of Figure 2. Results are shown in Figure 7.



■ **Figure 7** As in Figure 3, but for the exponent set A_{ex} and additional custom pattern families F^1 , F^2 , and F^3 as introduced in Example 26.

One observes that particular choices of pattern families, such as F^3 , may bring additional advantages with respect to runtime and approximation quality when compared to method H.

References

- 1 C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. A global optimization method, α BB, for general twice-differentiable constrained NLPs – II. Implementation and computational results. *Comput. Chem. Eng.*, 22(9):1159–1179, 1998.
- 2 Amir Ali Ahmadi, Cemil Dibek, and Georgina Hall. Sums of separable and quadratic polynomials. *Math. Oper. Res.*, 48(3):1316–1343, 2023.
- 3 Amir Ali Ahmadi and Anirudha Majumdar. Dsos and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM J. Appl. Algebra Geom.*, 3(2):193–230, 2019.
- 4 Martin Andersen, Joachim Dahl, Zhang Liu, and Lieven Vandenbergh. *Interior-point methods for large-scale cone programming*, chapter 3. Neural Information Processing series. MIT Press, 2011.
- 5 Miguel F. Anjos and Jean B. Lasserre, editors. *Handbook on semidefinite, conic and polynomial optimization*, volume 166 of *International Series in Operations Research & Management Science*. Springer, 2012.
- 6 Mosek ApS. Mosek modeling cookbook, 2020.
- 7 Johannes Aspman, Gilles Bareilles, Vyacheslav Kungurtsev, Jakub Marecek, and Martin Takáč. Hybrid methods in polynomial optimisation. <https://arxiv.org/abs/2305.16122>, 2023.
- 8 Gennadiy Averkov. Optimal size of linear matrix inequalities in semidefinite approaches to polynomial optimization. *SIAM J. Appl. Algebra Geom.*, 3(1):128–151, 2019.
- 9 Gennadiy Averkov and Claus Scheiderer. Convex hulls of monomial curves, and a sparse positivstellensatz. *Math. Program.*, 209(1-2):113–131, 2025.
- 10 Xiaowei Bao, Aida Khajavirad, Nikolaos V. Sahinidis, and Mohit Tawarmalani. Global optimization of nonconvex problems with multilinear intermediates. *Math. Program. Comput.*, 7(1):1–37, 2015.
- 11 P. Belotti. Couenne, an exact solver for nonconvex MINLPs, 2015. IBM and Carnegie Mellon University, <https://projects.coin-or.org/Couenne/>.
- 12 Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Society for Industrial and Applied Mathematics, 2001.
- 13 Grigoriy Blekherman, Pablo A. Parrilo, and Rekha R. Thomas, editors. *Semidefinite optimization and convex algebraic geometry*, volume 13 of *MOS-SIAM Series on Optimization*. Society for Industrial and Applied Mathematics; Mathematical Optimization Society, Philadelphia, PA, 2013.

- 14 Natasha Boland, Santanu S. Dey, Thomas Kalinowski, Marco Molinaro, and Fabian Rigterink. Bounding the gap between the McCormick relaxation and the convex hull for bilinear functions. *Math. Program.*, 162(1-2):523–535, 2017.
- 15 Venkat Chandrasekaran and Parikshit Shah. Relative entropy relaxations for signomial optimization. *SIAM J. Optim.*, 26(2):1147–1173, 2016.
- 16 Evrim Dalkiran and Hanif D. Sherali. Theoretical filtering of RLT bound-factor constraints for solving polynomial programming problems to global optimality. *J. Glob. Optim.*, 57(4):1147–1172, 2013.
- 17 Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *J. Mach. Learn. Res.*, 17: article no. 83 (5 pages), 2016.
- 18 H. Diedam and S. Sager. Global optimal control with the direct multiple shooting method. *Optim. Control Appl. Methods*, 39(2):449–470, 2018.
- 19 Mareike Dressler, Sadik Iliman, and Timo de Wolff. A Positivstellensatz for sums of nonnegative circuit polynomials. *SIAM J. Appl. Algebra Geom.*, 1(1):536–555, 2017.
- 20 W. R. Esposito and C. A. Floudas. Deterministic Global Optimization in Nonlinear Optimal Control Problems. *J. Glob. Optim.*, 17(1-4):97–126, 2000.
- 21 W. R. Esposito and C. A. Floudas. Global Optimization for the Parameter Estimation of Differential-Algebraic Systems. *Ind. Eng. Chem. Res.*, 39(5):1291–1310, 2000.
- 22 Hamza Fawzi. On representing the positive semidefinite cone using the second-order cone. *Math. Program.*, 175(1-2):109–118, 2019.
- 23 Michael Garstka, Mark Cannon, and Paul Goulart. COSMO: A conic operator splitting method for convex conic problems. *J. Optim. Theory Appl.*, 190(3):779–810, 2021.
- 24 Bernd Gärtner and Jiri Matousek. *Approximation algorithms and semidefinite programming*. Springer, 2012.
- 25 Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995.
- 26 Paul Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives. <https://arxiv.org/abs/2405.12762>, 2024.
- 27 Michael Grant and Stephen Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 2.1. <http://cvxr.com/cvx>, 2014.
- 28 Sander Gribling and Sven Polak. Mutually unbiased bases: polynomial optimization and symmetry. *Quantum*, 8:1318, 2024.
- 29 David Handelman. Representing polynomials by positive linear functions on compact convex polyhedra. *Pac. J. Math.*, 132(1):35–62, 1988.
- 30 Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. *Optim. Methods Softw.*, 24(4-5):761–779, 2009.
- 31 Marie Ioannou and Denis Rosset. Noncommutative polynomial optimization under symmetry. <https://arxiv.org/abs/2112.10803>, 2021.
- 32 Shucheng Kang, Guorui Liu, and Heng Yang. Global contact-rich planning with sparsity-rich semidefinite relaxations. <https://arxiv.org/abs/2502.02829>, 2025.
- 33 Shucheng Kang, Guorui Liu, and Heng Yang. Global contact-Rich planning with sparsity-rich semidefinite relaxations. In *Robotics: Science and Systems*, 2025. paper ID 46.
- 34 Lukas Katthän, Helen Naumann, and Thorsten Theobald. A unified framework of SAGE and SONC polynomials and its duality theory. *Math. Comput.*, 90(329):1297–1322, 2021.
- 35 A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28:497–520, 1960.
- 36 Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2001.
- 37 Jean B. Lasserre. *Moments, positive polynomials and their applications*, volume 1 of *Imperial College Press Optimization Series*. Imperial College Press, 2010.
- 38 Jean B. Lasserre. *An introduction to polynomial and semi-algebraic optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2015.
- 39 Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, volume 149 of *The IMA Volumes in Mathematics and its Applications*, pages 157–270. Springer, 2009.
- 40 Benoît Legat, Chris Coey, Robin Deits, Joey Huchette, and Amelia Perry. Sum-of-squares optimization in Julia. In *The First Annual JuMP-dev Workshop*, 2017.
- 41 John D. C. Little, Katta G. Murty, Dura W. Sweeney, and Caroline Karel. An Algorithm for the Traveling Salesman Problem. *Oper. Res.*, 11(6):972–989, 1963.
- 42 Johan Löfberg. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. In *Proceedings of the CACSD Conference*, pages 284–289, 2004.

- 43 Benoît Lubin, Oscar Dowson, Joaquim Garcia Garcia, Joey Huchette, Miles Lubin, and Juan P. Vielma. JuMP 1.0: recent improvements to a modeling language for mathematical optimization. *Math. Program. Comput.*, 15(3):581–589, 2023.
- 44 Victor Magron and Jie Wang. Tssos: a Julia library to exploit sparsity for large-scale polynomial optimization. <https://arxiv.org/abs/2103.00915>, 2021.
- 45 Victor Magron and Jie Wang. *Sparse polynomial optimization: theory and practice*. World Scientific, 2023.
- 46 Murray Marshall. *Positive polynomials and sums of squares*, volume 146 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2008.
- 47 A. Mitsos, B. Chachuat, and P. I. Barton. McCormick-Based Relaxations of Algorithms. *SIAM J. Optim.*, 20(2):573–601, 2009.
- 48 Hans D. Mittelmann and Frank Vallentin. High-accuracy semidefinite programming bounds for kissing numbers. *Exp. Math.*, 19(2):175–179, 2010.
- 49 R. E. Moore. *Interval analysis*. Prentice Hall, 1966.
- 50 MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual. Version 9.0., 2019. <http://docs.mosek.com/9.0/toolbox/index.html>.
- 51 Philippe Moustrou, Cordian Riener, and Hugues Verdure. Symmetries in polynomial optimization. In *Polynomial Optimization, Moments, and Applications*, pages 53–111. Springer, 2023.
- 52 A. Neumaier. Complete Search in Continuous Global Optimization and Constraint Satisfaction. *Acta Numer.*, 13:271–369, 2004.
- 53 Jiawang Nie. *Moment and Polynomial Optimization*. Society for Industrial and Applied Mathematics, 2023. MOS-SIAM Series on Optimization.
- 54 B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding. *J. Optim. Theory Appl.*, 169(3):1042–1068, 2016.
- 55 I. Papamichail and C. S. Adjiman. A Rigorous Global Optimization Algorithm for Problems with Ordinary Differential Equations. *J. Glob. Optim.*, 24(1):1–33, 2002.
- 56 B. Peters. *Monomial Patterns in Polynomial Optimization*. PhD thesis, Otto von Guericke University Magdeburg, 2021.
- 57 Stephen Prajna, Antonis Papachristodoulou, Peter Seiler, and Pablo A. Parrilo. New developments in sum of squares optimization and SOSTOOLS. In *Proceedings of the 2004 American control conference. Vol. 6*, pages 5606–5611. IEEE Press, 2004.
- 58 Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math. J.*, 42(3):969–984, 1993.
- 59 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254. ACM Press, 2008.
- 60 Cordian Riener, Thorsten Theobald, Lina Jansson Andrén, and Jean B. Lasserre. Exploiting symmetries in SDP-relaxations for polynomial optimization. *Math. Oper. Res.*, 38(1):122–141, 2013.
- 61 Nikolaos V. Sahinidis. BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs, Users Manual, 2017.
- 62 James Saunderson. Limitations on the expressive power of convex cones without long chains of faces. *SIAM J. Optim.*, 30(1):1033–1047, 2020.
- 63 Rolf Schneider. *Convex bodies: the Brunn-Minkowski theory*, volume 151 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, expanded edition, 2014.
- 64 Joseph K. Scott, Matthew D. Stuber, and Paul I. Barton. Generalized McCormick relaxations. *J. Glob. Optim.*, 51(4):569–606, 2011.
- 65 Henning Seidler and Timo de Wolff. An Experimental Comparison of SONC and SOS Certificates for Unconstrained Optimization. <https://arxiv.org/abs/1808.08431>, 2018.
- 66 Daria Shaydurova, Volker Kaibel, and Sebastian Sager. Refined TSSOS. *SIAM J. Optim.*, 35(2):1246–1273, 2025.
- 67 Hanif D. Sherali and Warren P. Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*, volume 31 of *Nonconvex Optimization and Its Applications*. Springer, 2013.
- 68 Edward M. B. Smith, Costas C. Pantelides, and Gintaras Reklaitis. A symbolic reformulation spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs. *Comput. Chem. Eng.*, 25:1399–1401, 2001.
- 69 Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11(1-4):625–653, 1999.
- 70 Ole Tange. Gnu Parallel 20201022, 2020. GNU Parallel is a general parallelizer to run multiple serial command line programs in parallel without changing them.
- 71 Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Math. Program.*, 103:225–249, 2005.
- 72 Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Math. Program.*, 103(2):225–249, 2005.

- 73 Mohit Tawarmalani and Nikolaos V. Sahinidis. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*, volume 65 of *Nonconvex Optimization and Its Applications*. Springer, 2013.
- 74 The Mathworks, Inc. MATLAB version 9.6.0.1174912 (R2019a) Update 5, 2019.
- 75 Kim-Chuan Toh, Michael J. Todd, and Reha H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 2.1. *Optim. Methods Softw.*, 11(1-4):545–581, 1999.
- 76 Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial and Applied Mathematics, 2019.
- 77 Lieven Vandenbergh, Martin Andersen, et al. Chordal graphs and semidefinite optimization. *Found. Trends Optim.*, 1(4):241–433, 2015.
- 78 Stefan Vigerske and Ambros Gleixner. SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optim. Methods Softw.*, 33(3):563–593, 2018.
- 79 Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.*, 17(1):218–242, 2006.
- 80 Hayato Waki, Sunyoung Kim, Masakazu Kojima, Masakazu Muramatsu, and Hiroshi Sugimoto. Algorithm 883: SparsePOP—a software package for polynomial optimization problems with structured sparsity. *ACM Trans. Math. Softw.*, 35(2):1–13, 2008.
- 81 Jie Wang, Victor Magron, and Jean-Bernard Lasserre. Chordal-TSSOS: A Moment-SOS Hierarchy That Exploits Term Sparsity with Chordal Extension. *SIAM J. Optim.*, 31(1):114–141, 2021.
- 82 Henry Wolkowicz, Romesh Saigal, and Lieven Vandenbergh. *Handbook of semidefinite programming: theory, algorithms, and applications*, volume 27 of *International Series in Operations Research & Management Science*. Springer, 2012.
- 83 Chenyang Yuan. SumOfSquares.py. <https://github.com/yuanchenyang/SumOfSquares.py>.
- 84 Yang Zheng, Giovanni Fantuzzi, and Antonis Papachristodoulou. Chordal and factor-width decompositions for scalable semidefinite and polynomial optimization. *Annu. Rev. Control*, 52:243–279, 2021.