SIAM J. OPTIM.
Vol. 29, No. 3, pp. 2128–2145

IN SDP RELAXATIONS, INACCURATE SOLVERS DO ROBUST OPTIMIZATION*

JEAN-BERNARD LASSERRE† AND VICTOR MAGRON†

Abstract. We interpret some incorrect results (due to numerical inaccuracies) already observed when solving semidefinite programming (SDP) relaxations for polynomial optimization on a double precision floating point SDP solver. It turns out that this behavior can be explained and justified satisfactorily by a relatively simple paradigm. In such a situation, the SDP solver, and not the user, performs some "robust optimization" without being told to do so. Instead of solving the original optimization problem with nominal criterion f, it uses a new criterion $\tilde{f}$ which belongs to a ball $\mathbf{B}_\infty(f,\varepsilon)$ of small radius $\varepsilon > 0$, centered at the nominal criterion f in the parameter space. In other words the resulting procedure can be viewed as a "max-min" robust optimization problem with two players (the solver which maximizes on $\mathbf{B}_\infty(f,\varepsilon)$ and the user who minimizes over the original decision variables). A mathematical rationale behind this "autonomous" behavior is described.

Key words. polynomial optimization, min-max optimization, robust optimization, semidefinite relaxations

AMS subject classification. 90C22

DOI. 10.1137/18M1225677

1. Introduction.

Certified optimization algorithms provide a way to ensure the safety of several systems in engineering sciences and program analysis, as well as cyber-physical critical components. Since these systems often involve nonlinear functions, such as polynomials, it is highly desirable to design certified polynomial optimization schemes and to be able to interpret the behaviors of numerical solvers implementing these schemes. Incorrect results (due to numerical inaccuracies) in some output results from semidefinite programming (SDP) solvers have been observed in quite different applications, and notably in recent applications of the Moment–Sum-of-squares (Moment-SOS) hierarchy for solving polynomial optimization problems; see, e.g., [21, 20]. In fact this particular application has even become a source of illustrative examples for potential pathological behavior of SDP solvers [17]. An intuitive mathematical rationale for the incorrect results has been already provided informally in [9] and [14], but does not yield a satisfactory picture for the whole process.

An immediate and irrefutable negative conclusion is that double precision floating point SDP solvers are not robust and cannot be trusted as they sometimes provide incorrect results in these so-called "pathological" cases. The present paper (with a voluntarily provocative title) is an attempt to provide a different and more positive viewpoint around the interpretation of such inaccuracies in SDP solvers, at least when applying the Moment-SOS hierarchy of semidefinite relaxations in polynomial optimization as described in [8, 10].

*Received by the editors November 7, 2018; accepted for publication (in revised form) June 7, 2019; published electronically August 27, 2019.
https://doi.org/10.1137/18M1225677
Funding: The work of the first author was supported by the European Research Council (ERC) under the European's Union Horizon 2020 research and innovation program grant 666981 TAMING. The work of the second author was supported by the FMJH program PGMO (EPICS project) and EDF, Thales, Orange et Criteo, as well as by the Tremplin ERC Stg grant ANR-18-ERC2-0004-01 (T-COPS project).
†CNRS, LAAS, F-31400 Toulouse, France (lasserre@laas.fr, https://homepages.laas.fr/lasserre/, vmagron@laas.fr, https://homepages.laas.fr/vmagron/).

We claim that in such a situation, in fact the floating point SDP solver, and not the user, is precisely doing some robust optimization, *without being told to do so.* It solves a "max-min" problem in a two-player zero-sum game where the solver is the leader who maximizes (over some ball of radius $\varepsilon > 0$) in the parameter space of the criterion, and the user is a "follower" who minimizes over the original decision variables. In traditional robust optimization, one solves the "min-max" problem where the user (now the leader) minimizes to find a "robust decision variable," whereas the SDP solver (now the follower) maximizes in the same ball of the parameter space. In this convex relaxation case, both min-max and max-min problems give the same solution. So it is fair to say that *the solver* is doing what the optimizer should have done in robust optimization.

As an active (and even leader) player of this game, the floating point SDP solver can also play with its two parameters, which are (a) the threshold level for eigenvalues to declare a matrix positive semidefinite, and (b) the tolerance level at which to declare a linear equality constraint to be satisfied. Indeed, the result of the "max-min" game strongly depends on the absolute value of both levels, as well as on their relative values.

Of course and so far, the rationale behind this viewpoint, which provides a more positive view of inaccurate results from semidefinite solvers, is we suited to the context of semidefinite relaxations for polynomial optimization. Indeed, in such a context we can exploit a mathematical rationale to explain and support this view. An interesting issue is to validate this viewpoint for a larger class of semidefinite programs and perhaps the canonical form of SDPs

$$\min_{\mathbf{X}} \{ \langle \mathbf{F}_0, \mathbf{X} \rangle : \langle \mathbf{F}_\alpha, \mathbf{X} \rangle = c_\alpha, \ \mathbf{X} \succeq 0 \},$$

in which case the SDP solver would solve the robust optimization problem

$$\max_{\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c},\varepsilon)} \min_{\mathbf{X}} \{ \langle \mathbf{F}_0, \mathbf{X} \rangle : \langle \mathbf{F}_\alpha, \mathbf{X} \rangle = \tilde{c}_\alpha, \ \mathbf{X} \succeq 0 \},$$

where $\langle \cdot \rangle$ stands for the matrix trace and "$\succeq 0$" means positive semidefinite. This point of view is briefly analyzed and discussed in section 3.4.

2. SDP solvers and the Moment-SOS hierarchy.

Notation. For a fixed $j \in \mathbb{N}$, let us denote by $\mathbb{R}[\mathbf{x}]_{2j}$ the set of polynomials of degree at most $2j$ and by $\mathcal{S}_{n,j}$ the set of real symmetric matrices of size $\binom{n+j}{n}$. For any real symmetric matrix $\mathbf{M}$, denote by $\|\mathbf{M}\|_*$ its *nuclear* norm and recall that if $\mathbf{M} \succeq 0$ then $\|\mathbf{M}\|_* = \langle \mathbf{I}, \mathbf{M} \rangle$. We also denote by $\Sigma[\mathbf{x}]_j$ the convex cone of SOS polynomials of degree at most $2j$. Let $\mathbb{N}^n_{2d} := \{(\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n : \alpha_1 + \cdots + \alpha_n \leq 2d\}$. In what follows, we will use a generalization of Von Neumann's minimax theorem, namely the following Sion minimax theorem [18].

THEOREM 2.1. *Let* $\mathbf{B}$ *be a compact convex subset of a linear topological space and* $\mathbf{Y}$ *be a convex subset of a linear topological space. If* h *is a real-valued function on* $\mathbf{B} \times \mathbf{Y}$ *with* $h(\mathbf{b}, \cdot)$ *lower semicontinuous and quasi-convex on* $\mathbf{Y}$ *for all* $\mathbf{b} \in \mathbf{B}$*, and* $h(\cdot, \mathbf{y})$ *upper semicontinuous and quasi-concave on* $\mathbf{B}$ *for all* $\mathbf{y} \in \mathbf{Y}$*, then*

$$\max_{\mathbf{b} \in \mathbf{B}} \inf_{\mathbf{y} \in \mathbf{Y}} h(\mathbf{b}, \mathbf{y}) = \inf_{\mathbf{y} \in \mathbf{Y}} \max_{\mathbf{b} \in \mathbf{B}} h(\mathbf{b}, \mathbf{y}).$$

The Moment-SOS hierarchy was introduced in [8] to solve the global polynomial optimization problem

$$\mathbf{P}: \quad f^\star = \min_{\mathbf{x}} \{ f(\mathbf{x}) : \mathbf{x} \in \mathbf{K} \},$$

where f is a polynomial and

$$\mathbf{K} := \{\mathbf{x} \in \mathbb{R}^n : \; g_l(\mathbf{x}) \geq 0, \; l = 1, \ldots, m\}$$

is a basic closed semialgebraic set with $(g_\ell) \subset \mathbb{R}[\mathbf{x}]$. Let us note that $g_0 := 1$ and $d_\ell := \deg g_\ell$ for each $\ell = 0, \ldots, m$.

A systematic numerical scheme consists of solving a hierarchy of convex relaxations

$$\text{(2.1)} \qquad \mathbf{P}^j : \quad \rho^j \; = \; \min_{\mathbf{y}} \{ \, L_{\mathbf{y}}(f) : \; y_0 = 1, \; \mathbf{y} \in C_j(g_1, \ldots, g_m) \, \} \, ,$$

where $(C_j(g_1, \ldots, g_m))_{j \in \mathbb{N}}$ is an appropriate nested family of convex cones, such as the one given in (3.3). The dual of (2.1) reads

$$\text{(2.2)} \qquad \mathbf{D}^j : \quad \delta^j \; = \; \max_{\lambda} \{ \, \lambda : \; f - \lambda \in C_j(g_1, \ldots, g_m)^\star \, \},$$

where $(C_j(g_1, \ldots, g_m)^\star)_{j \in \mathbb{N}} \subset \mathbb{R}[\mathbf{x}]$ is a nested family of convex cones contained in $C(\mathbf{K})$, the convex cone of polynomials nonnegative on $\mathbf{K}$, and $L_{\mathbf{y}} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ is the Riesz linear functional:

$$f \quad \left(= \sum_\alpha f_\alpha \, x^\alpha \right) \mapsto L_{\mathbf{y}}(f) = \sum_\alpha f_\alpha \, y_\alpha \, .$$

When $C_j(g_1, \ldots, g_m)^\star$ comes from an appropriate SOS-based (Putinar) representation of polynomials positive on $\mathbf{K}$, both $\mathbf{P}^j$ and $\mathbf{D}^j$ are *semidefinite programs*. When $\mathbf{K}$ is compact then (under a weak archimedean condition) $\rho^j = \delta^j \uparrow f^\star$ as $j \to \infty$, and generically the convergence is even finite [16], i.e., $f^\star = \rho^j$ for some $j \in \mathbb{N}$. In such a case, one may also extract global minimizers from an optimal solution of the corresponding semidefinite relaxation $\mathbf{P}^j$ [15]. For more details on the Moment-SOS hierarchy, the interested reader is referred to [10].

At step j in the hierarchy, one has to solve the SDP-relaxation $\mathbf{P}^j$, for which efficient modern software is available. These numerical solvers all rely on interior-point methods, and are implemented either in double precision arithmetic (e.g., SeDuMi [19], SDPA [22], Mosek [1]) or with arbitrary precision arithmetic (e.g., SDPA-GMP [13]). When relying on such numerical frameworks, the input data considered by solvers might differ from the ones given by the user. Thus the input data, consisting of the cost vector and matrices, are subject to uncertainties. In [4] the authors study semidefinite programs whose input data depend on some unknown but bounded perturbation parameters. For the reader interested in robust optimization in general, we refer them to [2].

2.1. Two examples of surprising phenomena. In general, when applied to solving $\mathbf{P}$, the Moment-SOS hierarchy [8] is quite efficient, despite its scalability (indeed for large size problems one has to exploit sparsity often encountered in the description of $\mathbf{P}$). However, in some cases, some quite surprising phenomena have been observed and provided additional support to the pessimistic and irrefutable conclusion that *results returned by double precision floating point SDP solvers cannot be trusted as they are sometimes completely wrong.*

Let us briefly describe two such phenomena, already analyzed and commented on in [21, 14].

Case 1. When $\mathbf{K} = \mathbb{R}^n$ (unconstrained optimization) the Moment-SOS hierarchy collapses to the single SDP $\rho^d = \max_\lambda \{ \lambda : f - \lambda \in \Sigma[\mathbf{x}]_d \}$ (with $2d$ being the degree of f). Equivalently, one solves the semidefinite program

$$\mathbf{D}^d : \quad \rho^d = \max_{\mathbf{X} \succeq 0, \lambda} \{ \lambda : f_\alpha - \lambda 1_{\alpha=0} = \langle \mathbf{X}, \mathbf{B}_\alpha \rangle, \ \alpha \in \mathbb{N}^n_{2d} \} \tag{2.3}$$

for some appropriate real symmetric matrices $(\mathbf{B}_\alpha)_{\alpha \in \mathbb{N}^n_{2d}}$; see, e.g., [8].

Only two cases can happen: if $f - f^* \in \Sigma[\mathbf{x}]_d$ then $\rho^d = f^*$, and $\rho^d < f^*$ otherwise (with possibly $\rho^d = -\infty$). Solving $\rho^j = \max_\lambda \{ \lambda : f - \lambda \in \Sigma[\mathbf{x}]_j \}$ for $j > d$ is useless as it would yield $\rho^j = \rho^d$, because if $f - f^*$ is SOS then it has to be in $\Sigma[\mathbf{x}]_d \subset \Sigma[\mathbf{x}]_j$ anyway.

The Motzkin-like polynomial $\mathbf{x} \mapsto f(\mathbf{x}) = x^2y^2(x^2+y^2-1)+1/27$ is nonnegative (with $d = 3$ and $f^* = 0$) and has 4 global minimizers, but the polynomial $\mathbf{x} \mapsto f(\mathbf{x}) - f^* (= f)$ is *not* an SOS and $\rho^3 = -\infty$, which also implies $\rho^j = -\infty$ for all j. However, as already observed in [5], by solving (2.3) with $j = 8$ and a double precision floating point SDP solver, we obtain $\rho_8 \approx -10^{-4}$. In addition, one may extract 4 global minimizers close to the global minimizers of f up to 4 digits of precision! The same occurs with $j > 8$ and the higher j is, the better the result. So undoubtedly the SDP solver is returning a wrong solution as $f - \rho^j$ *cannot* be an SOS, no matter the value of ρ^j.

In this case, a rationale for this behavior is that $\tilde{f} = f + \varepsilon(1 + x^{16} + y^{16})$ is an SOS for small $\varepsilon > 0$, provided that ε is not too small (in [9] it is shown that every nonnegative polynomial can be approximated as closely as desired by a sequence of polynomials that are sums of squares). After inspection of the returned optimal solution, the equality constraints

$$f_\alpha - \lambda 1_{\alpha=0} = \langle \mathbf{X}, \mathbf{B}_\alpha \rangle, \quad \alpha \in \mathbb{N}^n_{2j}, \tag{2.4}$$

when solving $\mathbf{D}^j$ in (2.3), are not satisfied accurately and the result can be interpreted as if the SDP solver *has replaced* f with the perturbed criterion $\tilde{f} = f + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon}(\mathbf{x}) = \sum_\alpha \varepsilon_\alpha \mathbf{X}^\alpha \in \mathbb{R}[\mathbf{x}]_{2d}$, so that

$$\underbrace{f_\alpha + \varepsilon_\alpha}_{\tilde{f}_\alpha} - \lambda 1_{\alpha=0} = \langle \mathbf{X}, \mathbf{B}_\alpha \rangle, \quad \alpha \in \mathbb{N}^n_{2j},$$

and in fact it has done so. A similar "mathematical paradox" has also been investigated in a noncommutative (NC) context [14]. NC polynomials can also be analyzed thanks to an NC variant of the Moment-SOS hierarchy (see [3] for a recent survey). As in the above commutative case, it is explained in [14] how numerical inaccuracies allow one to obtain converging lower bounds for positive Weyl polynomials that do not admit SOS decompositions.

Case 2. Another surprising phenomenon occurred when minimizing a high-degree univariate polynomial f with a global minimizer at $x = 100$ and a local minimizer at $x = 1$ with value $f(1) > f^*$ but very close to $f^* = f(100)$. The double precision floating point SDP solver returns a single minimizer $\tilde{x} \approx 1$ with value very close to f^*, providing another irrefutable proof that the double precision floating point SDP solver has returned an incorrect solution. It turns out that again the result can be interpreted as if the SDP solver *has replaced* f with a perturbed criterion $\tilde{f}$, as in Case 1.

When solving (2.3) in Case 1, one has voluntarily embedded $f \in \mathbb{R}[\mathbf{x}]_6$ into $\mathbb{R}[\mathbf{x}]_{2j}$ (with $j > 3$) to obtain a perturbation $\tilde{f} \in \mathbb{R}[\mathbf{x}]_{2j}$ whose minimizers are close enough to those of f. Of course the precision is in accordance with the solver parameters involved in controlling the semidefiniteness of the moment matrix $\mathbf{X}$ and the accuracy of the linear equations (2.4). Indeed, if one tunes these parameters to a much stronger threshold, then the solver returns a more accurate answer with a much higher precision.

In both contexts, we can interpret what the SDP solver does as perturbing the coefficients of the input polynomial data. One approach to get rid of numerical uncertainties consists of solving SDP problems in an exact way [6] while using symbolic computation algorithms. However, such exact algorithms only scale up to moderate size instances. For situations when one has to rely on more efficient, yet inexact, numerical algorithms, there is a need to understand the behavior of the associated numerical solvers. In [21], the authors investigate strange behaviors of double-precision SDP solvers for semidefinite relaxations in polynomial optimization. They compute the optimal values of the SDP relaxations of a simple one-dimensional polynomial optimization problem. The sequence of SDP values practically converges to the optimal value of the initial problem while they should converge to a strict lower bound of this value. One possible remedy, used in [21], is to rely on an arbitrary-precision SDP solver, such as SDPA-GMP [13], in order to make this paradoxical phenomenon disappear. Relying on such arbitrary-precision solvers comes together with a more expensive cost but paves a way towards exact certification of nonnegativity. In [11], the authors present a hybrid numeric-symbolic algorithm computing exact SOS certificates for a polynomial lying in the interior of the SOS cone. This algorithm uses SDP solvers to compute an approximate SOS decomposition after additional perturbation of the coefficients of the input polynomial. The idea is to benefit from the perturbation terms added by the *user* to compensate the numerical uncertainties added by the *solver*. The present note focuses on analyzing specifically how the solver modifies the input and perturbs the polynomials of the initial optimization problem.

2.2. Contribution. We claim that there is also another possible and more optimistic conclusion if one looks at the above results with new "robust optimization" glasses, not from the viewpoint of the user but rather from the view point of the solver. More precisely, given a polynomial optimization problem $f^\star = \min_{\mathbf{x}}\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\}$ and its semidefinite relaxation $\mathbf{P}^j$ defined in (2.1) (with dual $\mathbf{D}^j$ in (2.2)):

> *We interpret the above behavior as the (double precision floating point) SDP solver doing "robust optimization" without being told to do so. In the case of individual trace equality perturbations ε, it solves the max-min problem*
>
> $$(2.5)\quad \rho^j_\varepsilon = \max_{\tilde f \in \mathbf{B}^j_\infty(f,\varepsilon)} \Big\{ \inf_{\mathbf{y}} \{ L_{\mathbf{y}}(\tilde f) : y_0 = 1,\ \mathbf{y} \in C_j(g_1,\dots,g_m) \} \Big\},$$
>
> *where* $\mathbf{B}^j_\infty(f,\varepsilon) := \{ \tilde f \in \mathbb{R}[\mathbf{x}]_{2j} : \|\tilde f - f\|_\infty \le \varepsilon \}$.

We also provide some numerical experiments to support this claim. Interestingly, if *the user* does robust optimization, then *he* solves the min-max problem

$$\inf_{\mathbf{y}} \Big\{ \max_{\tilde f \in \mathbf{B}^j_\infty(f,\varepsilon)} \{ L_{\mathbf{y}}(\tilde f) \},\ y_0 = 1,\ \mathbf{y} \in C_j(g_1,\dots g_m) \Big\}, \tag{2.6}$$

which is (2.5) but where the "max" and "min" operators have been switched. It turns out that in this convex case, by Theorem 2.1, the optimal value of (2.6) is ρ^j_ε.

So from a *robustness* view point of the solver (not the user), it is quite reasonable to solve (2.5) rather than the original relaxation $\mathbf{P}^j$ of $\mathbf{P}$ with nominal polynomial f.

However, since ρ_ε^j is equal to the optimal value of (2.6), the result is the same as if the user decided to do "robust optimization"! In other words, solving $\mathbf{P}^j$ with nominal f and numerical inaccuracies is the same as solving the robust problems (2.5) or (2.6) with infinite precision.

3. A "noise" model.

Given a finite sequence of matrices $(\mathbf{F}_\alpha)_{\alpha\in\mathbb{N}^n_{2j}} \subset \mathcal{S}_{n,j}$ and a (primal) cost vector $\mathbf{c} = (c_\alpha)_{\alpha\in\mathbb{N}^n_{2j}}$, we recall the standard form of a *primal* semidefinite program solved by numerical solvers such as SDPA [22], that being

$$\begin{aligned} \min_{\mathbf{y}} \quad & \sum_{\alpha\in\mathbb{N}^n_{2j}} c_\alpha\, y_\alpha \\ \text{s.t.} \quad & \sum_{0\neq\alpha\in\mathbb{N}^n_{2j}} \mathbf{F}_\alpha\, y_\alpha \succeq \mathbf{F}_0\,, \end{aligned} \tag{3.1}$$

whose *dual* is the following SDP optimization problem:

$$\begin{aligned} \max_{\mathbf{X}} \quad & \langle \mathbf{F}_0, \mathbf{X}\rangle \\ \text{s.t.} \quad & \langle \mathbf{F}_\alpha, \mathbf{X}\rangle = c_\alpha\,, \quad \alpha\in\mathbb{N}^n_{2j}\,, \quad \alpha\neq 0\,, \\ & \mathbf{X}\succeq 0\,, \quad \mathbf{X}\in\mathcal{S}_{n,j}\,. \end{aligned} \tag{3.2}$$

We are interested in the numerical analysis of the moment-SOS hierarchy [8] to solve

$$\mathbf{P}: \quad \min_{\mathbf{x}\in\mathbf{K}} f(\mathbf{x})\,,$$

where $f \in \mathbb{R}[\mathbf{x}]_{2j}$. Given $\alpha, \beta \in \mathbb{N}^n$, let $1_{\alpha=\beta}$ stand for the function which returns 1 if $\alpha = \beta$ and 0 otherwise. At step d of the hierarchy, one solves the SDP primal program (2.1). For the standard choice of the convex cone $C_j(g_1,\dots,g_m)$, given by

$$C_j(g_1,\dots,g_m) = \{\mathbf{y} : \mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y}) \succeq 0,\ \ell = 0,\dots,m\}\,, \tag{3.3}$$

program (2.1) reads

$$\rho^j = \inf_{\mathbf{y}} \{\, L_{\mathbf{y}}(f) :\ y_0 = 1;\ \mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y}) \succeq 0,\ \ell = 0,\dots,m\,\}\,, \tag{3.4}$$

whose dual is the SDP

$$\begin{aligned} \delta^j = \sup_{\mathbf{X}_\ell,\lambda} \Bigg\{ \lambda :\ & f_\alpha - \lambda\, 1_{\alpha=0} = \sum_{\ell=0}^m \langle \mathbf{C}^\ell_\alpha, \mathbf{X}_\ell\rangle\,,\ \alpha\in\mathbb{N}^n_{2j}\,, \\ & \mathbf{X}_\ell \succeq 0\,,\ \mathbf{X}_\ell \in \mathbf{S}_{n,j-d_\ell}\,,\ \ell = 0,\dots,m \Bigg\}, \end{aligned} \tag{3.5}$$

where we have written $\mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y}) = \sum_{\alpha\in\mathbb{N}^n_{2j}} \mathbf{C}^\ell_\alpha\, y_\alpha$; the matrix $\mathbf{C}^\ell_\alpha$ has rows and columns indexed by $\mathbb{N}^n_{j-d_\ell}$ with the (β,γ) entry equal to $\sum_{\beta+\gamma+\delta=\alpha} g_{\ell,\delta}$. In particular, for $m = 0$ one has $g_0 = 1$ and the matrix $\mathbf{B}_\alpha := \mathbf{C}^0_\alpha$ has (β,γ) entry equal to $1_{\beta+\gamma=\alpha}$.

For every $j \in \mathbb{N}$, let

$$\mathcal{Q}_j(g) \;=\; \left\{ \sum_{\ell=0}^m \sigma_\ell\, g_\ell :\ \deg(\sigma_\ell\, g_\ell)\ \leq 2j\,,\ \sigma_\ell \in \Sigma[\mathbf{x}] \right\}$$

be the "truncated" quadratic module associated with the g_ℓ's.

Then the dual SDP (3.5) can be rewritten as

$$\begin{aligned}\delta^j &= \sup_{\lambda} \{\lambda : f - \lambda \in \mathcal{Q}_j(g)\} \\ &= \sup_{\lambda,\sigma_\ell} \left\{ \lambda : \ f - \lambda = \sum_{\ell=0}^{m} \sigma_\ell \, g_\ell \,, \ \deg(\sigma_\ell \, g_\ell) \le 2j \,, \ \sigma_\ell \in \Sigma[\mathbf{x}] \right\}.\end{aligned} \tag{3.6}$$

Strong duality of Lasserre's hierarchy is guaranteed when the following condition (slightly stronger than compactness of $\mathbf{K}$) holds.

Assumption 3.1. There exists $N \in \mathbb{N}$ such that one of the polynomials describing the set $\mathbf{K}$ reads $g^{\mathbf{K}}(\mathbf{x}) := N - \|\mathbf{x}\|_2^2$.

Then it follows from [7] that this ball constraint implies strong duality between (3.4) and (3.6). Note that if the set $\mathbf{K}$ is bounded, then one can add the redundant constraint $N - \|\mathbf{x}\|_2^2 \ge 0$ without modifying $\mathbf{K}$. In what follows, we suppose that Assumption 3.1 holds.

In floating point computation, the numerical SDP solver treats all (ideally) equality constraints

$$\sum_{\ell=0}^{m} \langle \mathbf{C}_\alpha^\ell, \mathbf{X}_\ell \rangle + \lambda 1_{\alpha=0} - f_\alpha = 0 \,, \quad \alpha \in \mathbb{N}_{2j}^n \,, \tag{3.7}$$

of (3.5) as the inequality constraints

$$\left| \sum_{\ell=0}^{m} \langle \mathbf{C}_\alpha^\ell, \mathbf{X}_\ell \rangle + \lambda 1_{\alpha=0} - f_\alpha \right| \le \varepsilon \,, \quad \alpha \in \mathbb{N}_{2j}^n, \tag{3.8}$$

for some a priori fixed tolerance $\varepsilon > 0$ (for instance $\varepsilon = 10^{-8}$). Similarly, we assume that for each $\ell = 0, \ldots, m$, the SDP constraint $\mathbf{X}_\ell \succeq 0$ of (3.5) is relaxed to $\mathbf{X}_\ell \succeq -\eta \, \mathbf{I}$ for some prescribed *individual semidefiniteness tolerance* $\eta > 0$. This latter relaxation of $\succeq 0$ to $\succeq -\eta\mathbf{I}$ is used here as an idealized situation for modeling purposes; in practice it seems to be more complicated, as explained later on at the beginning of section 4.

That is, all iterates $(\mathbf{X}_{\ell,k})_{k\in\mathbb{N}}$ of the implemented minimization algorithm satisfy (3.8) and $\mathbf{X}_{\ell,k} \succeq -\eta\mathbf{I}$ instead of the idealized (3.7) and $\mathbf{X}_{\ell,k} \succeq 0$.

Therefore we interpret the SDP solver behavior by considering the following "noise" model, which is the (ε, η)-perturbed version of SDP (3.5):

$$\begin{aligned}\sup_{\mathbf{X}_\ell, \lambda} \Bigg\{ \lambda : \ -\varepsilon \le \sum_{\ell=0}^{m} \langle \mathbf{C}_\alpha^\ell, \mathbf{X}_\ell \rangle + \lambda 1_{\alpha=0} - f_\alpha \le \varepsilon \,, \ \alpha \in \mathbb{N}_{2j}^n \,, \\ \mathbf{X}_\ell \succeq -\eta \, \mathbf{I} \,, \ \mathbf{X}_\ell \in \mathbf{S}_{n, j-d_\ell} \,, \ \ell = 0, \ldots, m \Bigg\},\end{aligned} \tag{3.9}$$

where we now assume exact computations.

PROPOSITION 3.2. *The dual of problem* (3.9) *is the convex optimization problem*

$$\begin{aligned}\inf_{\mathbf{y}} \Bigg\{ L_{\mathbf{y}}(f) + \eta \sum_{\ell=0}^{m} \|\mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y})\|_* + \varepsilon \|\mathbf{y}\|_1 : \\ y_0 = 1; \ \mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y}) \succeq 0, \ \ell = 0, \ldots, m \Bigg\},\end{aligned} \tag{3.10}$$

which is an SDP.

Proof. Let $y_\alpha^\pm$ be the nonnegative dual variables associated with the constraints

$$\pm \left(\sum_{\ell=0}^{m} \langle \mathbf{C}_\alpha^\ell, \mathbf{X}_\ell \rangle + \lambda 1_{\alpha=0} - f_\alpha \right) \le \varepsilon, \quad \alpha \in \mathbb{N}_{2j}^n,$$

and let $\mathbf{S}_\ell \succeq 0$ be the dual matrix variable associated with the SDP constraint $\mathbf{X}_\ell \succeq -\eta\, \mathbf{I}$, $\ell = 0, \ldots, m$. Then the dual of (3.9) is a semidefinite program which reads

$$\begin{aligned} \inf_{\mathbf{S}_\ell \succeq 0, y_\alpha^\pm \ge 0} \Bigg\{ & \sum_\alpha (f_\alpha\,(y_\alpha^+ - y_\alpha^-) + \varepsilon\,(y_\alpha^+ + y_\alpha^-)) + \eta \sum_\ell \langle \mathbf{I}, \mathbf{S}_\ell \rangle : \\ & \mathbf{S}_\ell - \sum_\alpha \mathbf{C}_\alpha^\ell\,(y_\alpha^+ - y_\alpha^-) \,=\, 0,\ \ell = 0, \ldots, m\,, \\ & y_0^+ - y_0^- \,= 1 \Bigg\}. \end{aligned} \tag{3.11}$$

In view of the nonnegative terms $\varepsilon \sum_\alpha (y_\alpha^+ + y_\alpha^-)$ in the criterion, at an optimal solution we necessarily have $y_\alpha^+ y_\alpha^- = 0$ for all α. Therefore, letting $y_\alpha := y_\alpha^+ - y_\alpha^-$, one obtains $y_\alpha^+ + y_\alpha^- = |y_\alpha|$ for all α, and $\sum_\alpha (y_\alpha^+ + y_\alpha^-) = \|\mathbf{y}\|_1$. Similarly, as $\mathbf{S}_\ell \succeq 0$, $\langle \mathbf{I}, \mathbf{S}_\ell \rangle = \|\mathbf{S}_\ell\|_1$, $\ell = 0, \ldots, m$. This yields the formulation (3.10). ∎

Remark 3.3. Notice that the criterion of (3.10) consists of the original criterion $L_\mathbf{y}(f)$ perturbed by a sparsity-inducing norm $\varepsilon\,\|\mathbf{y}\|_1$ for the variable $\mathbf{y}$ and a low-rank-inducing norm $\eta \sum_\ell \|\mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y})\|_*$ for the localizing matrices. Note that this low-rank-inducing term can be seen as the convexification of a more realistic penalization with a logarithmic barrier function used in interior-point methods for SDP, namely $-\eta \log\det(\sum_{\ell=0}^m \mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y}))$. One could also consider replacing each SDP constraint $\mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y}) \succeq 0$ with $\mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y}) \succeq \varepsilon_3 \mathbf{I}$ in the primal moment problem (3.4). This corresponds to adding $-\varepsilon_3 \|\mathbf{X}\|_*$ in the related perturbation of the dual SOS problem (3.5). One can in turn interpret this term as a convexification of the more standard logarithmic barrier penalization term $\log\det \mathbf{X}$. Even though interior-point algorithms could practically perform such logarithmic barrier penalizations, we do not have a simple interpretation for the related noise model.

We now distinguish between two particular cases.

3.1. Priority to trace equalities.

With $\varepsilon = 0$ and individual semidefiniteness tolerance η, problem (3.10) becomes

$$\begin{aligned} \rho_\eta^j = \ \inf_{\mathbf{y}} \Bigg\{ & L_\mathbf{y}(f) + \eta \sum_{\ell=0}^m \|\mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y})\|_* \\ & \text{s.t. } y_0 = 1;\ \mathbf{M}_{j-d_\ell}(g_\ell\,\mathbf{y}) \succeq 0,\ \ell = 0, \ldots, m \Bigg\}. \end{aligned} \tag{3.12}$$

Given $\eta > 0$, $j \in \mathbb{N}$, let us define

$$\begin{aligned} \mathbf{B}_\infty^j(f, \mathbf{K}, \eta) &:= \left\{ f + \theta \sum_{\ell=0}^m g_\ell(\mathbf{x}) \sum_{\beta \in \mathbb{N}_{j-d_\ell}^n} \mathbf{x}^{2\beta} : |\theta| \le \eta \right\}, \\ \mathbf{B}_\infty(f, \mathbf{K}, \eta) &:= \bigcup_{j \in \mathbb{N}} \mathbf{B}_\infty^j(f, \mathbf{K}, \eta)\,. \end{aligned} \tag{3.13}$$

Recall that SDP (3.12) is the dual of SDP (3.9) with $\varepsilon = 0$, that is,

$$\sup_{\mathbf{X}_\ell,\lambda} \left\{ \lambda : \; f_\alpha - \lambda 1_{\alpha=0} = \sum_{\ell=0}^{m} \langle \mathbf{C}_\alpha^\ell, \mathbf{X}_\ell \rangle \,, \; \alpha \in \mathbb{N}^n_{2j} \,, \right. \tag{3.14}$$
$$\left. \mathbf{X}_\ell \succeq -\eta \, \mathbf{I} \,, \; \mathbf{X}_\ell \in \mathbf{S}_{n,j-d_\ell} \,, \; \ell = 0, \ldots, m \right\}.$$

Fix $j \in \mathbb{N}$ and consider the following robust polynomial optimization problem:

$$\mathbf{P}_\eta^{\max} : \quad \max_{\tilde{f} \in \mathbf{B}_\infty(f,\mathbf{K},\eta)} \left\{ \min_{\mathbf{x} \in \mathbf{K}} \{ \tilde{f}(\mathbf{x}) \} \right\}. \tag{3.15}$$

If in (3.15) we restrict ourselves to $\mathbf{B}^j_\infty(f,\mathbf{K},\eta)$ and we replace the inner minimization by its step-j relaxation, we obtain

$$\mathbf{P}_\eta^{\max,j} : \quad \max_{\tilde{f} \in \mathbf{B}^j_\infty(f,\mathbf{K},\eta)} \left\{ \inf_{\mathbf{y}} \{ L_\mathbf{y}(\tilde{f}) : \; y_0 = 1; \; \mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y}) \succeq 0, \; \ell = 0, \ldots, m \} \right\}.$$

Observe that problem $\mathbf{P}_\eta^{\max,j}$ is a strengthening of problem $\mathbf{P}_\eta^{\max}$, that is, the optimal value of the former is smaller than the optimal value of the latter.

PROPOSITION 3.4. *Under Assumption* 3.1*, there is no duality gap between primal SDP* (3.12) *and dual SDP* (3.14)*. In addition, problem* $\mathbf{P}_\eta^{\max,j}$ *is equivalent to SDP* (3.12)*. Therefore, solving primal SDP* (3.12) *or dual SDP* (3.14) *can be interpreted as solving* exactly*, i.e., with no semidefiniteness-tolerance, the step-j strengthening* $\mathbf{P}_\eta^{\max,j}$ *associated with problem* $\mathbf{P}_\eta^{\max}$*.*

Proof. Recall that for every $\ell = 0, \ldots, m$, one has $\mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y}) \succeq 0$ and

$$\|\mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y})\|_* \, = \, \mathrm{Trace}\,(\mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y})) = L_\mathbf{y}\Bigg(\sum_{\beta \in \mathbb{N}^n_{j-d_\ell}} \mathbf{x}^{2\beta} \, g_\ell(\mathbf{x}) \Bigg).$$

For $\tilde{f} = f + \eta \sum_{\beta \in \mathbb{N}^n_{j-d_\ell}} \mathbf{x}^{2\beta} \, g_\ell(\mathbf{x})$, one has

$$L_\mathbf{y}(\tilde{f}) = L_\mathbf{y}(f) + \eta \sum_{\ell=0}^{m} \|\mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y})\|_* \,.$$

Thus, the primal SDP (3.12) (resp., dual SDP (3.14)) boils down to solving the primal SDP (3.4) (resp., (3.5)) after replacing f by $\tilde{f}$. By Assumption 3.1, there is no duality gap between (3.4) and (3.5), and thus there is also no duality gap between (3.12) and (3.14).

In addition, since $\tilde{f}$ is feasible for problem $\mathbf{P}_\eta^{\max,j}$, the optimal value of problem $\mathbf{P}_\eta^{\max,j}$ is greater than the value of SDP (3.12). By Theorem 2.1, problem $\mathbf{P}_\eta^{\max,j}$ is equivalent to

$$\inf_{\mathbf{y}} \max_{\tilde{f} \in \mathbf{B}^j_\infty(f,\mathbf{K},\eta)} \{ L_\mathbf{y}(\tilde{f}) : y_0 = 1; \; \mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y}) \succeq 0, \; \ell = 0, \ldots, m \}. \tag{3.16}$$

For all $\tilde{f} \in \mathbf{B}^j_\infty(f,\mathbf{K},\eta)$, $L_\mathbf{y}(\tilde{f}) \leq L_\mathbf{y}(f) + \eta \sum_{\ell=0}^m \|\mathbf{M}_{j-d_\ell}(g_\ell \, \mathbf{y})\|_*$, which proves that the optimal value of (3.16) is less than the value of SDP (3.12).

This yields the equivalence between problem $\mathbf{P}_\eta^{\max,j}$ and SDP (3.12). $\square$

In the unconstrained case, i.e., when $m = 0$, solving $\mathbf{P}_\eta^{\max,j}$ boils down to minimizing the perturbed polynomial

$$f_{\eta,j}(\mathbf{x}) := f(\mathbf{x}) + \eta \sum_{|\beta| \leq j} \mathbf{x}^{2\beta},$$

that is, the sum of f and all monomial squares of degree up to $2j$ with coefficient magnitude η. As a direct consequence of [9], the next result shows that for given nonnegative polynomial f and perturbation $\eta > 0$, the polynomial $f_{\eta,j}$ is SOS for large enough j.

COROLLARY 3.5. *Let us assume that $f \in \mathbb{R}[\mathbf{x}]$ is nonnegative over $\mathbb{R}^n$ and let us fix $\eta > 0$. Then $f_{\eta,j} \in \Sigma[\mathbf{x}]$ for large enough j.*

Proof. For fixed nonnegative $f \in \mathbb{R}[\mathbf{x}]$ and $\eta > 0$, it follows from [9, Theorem 4.2(ii)] that there exists j_η (depending on f and η) such that the polynomial

$$f + \eta \sum_{k=0}^{j} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!}$$

is SOS for any $d \geq d_\eta$. Let us select $j := j_\eta$. Notice that

$$f_{\eta,j} = f + \eta \sum_{|\beta| \leq j} \mathbf{x}^{2\beta} = f + \eta \sum_{k=0}^{j} \sum_{i=1}^{n} \frac{x_i^{2k}}{k!} + \eta \sum_{k=0}^{j} \sum_{i=1}^{n} \left(1 - \frac{1}{k!}\right) x_i^{2k} + \eta\, q_j,$$

where q_j is a sum of monomial squares. Since $(1 - \frac{1}{k!}) \geq 0$, the second sum of the right-hand side is SOS, yielding the desired claim. ∎

3.2. Priority to semidefiniteness inequalities.

3.2. Priority to semidefiniteness inequalities. Problem (3.10) with $\eta = 0$ and individual trace equality perturbation ε becomes

$$\begin{aligned} \rho_\varepsilon^j = \inf_{\mathbf{y}} \{\, & L_{\mathbf{y}}(f) + \varepsilon \, \|\mathbf{y}\|_1 \ : \\ & \text{s.t. } y_0 = 1;\ \mathbf{M}_{j-d_\ell}(g_\ell\, \mathbf{y}) \succeq 0,\ \ell = 0, \ldots, m \,\}. \end{aligned} \tag{3.17}$$

Given $\varepsilon > 0$, $j \in \mathbb{N}$, let us define

$$\mathbf{B}_\infty^j(f, \varepsilon) := \{\, \tilde{f} \in \mathbb{R}[\mathbf{x}]_{2j} : \|f - \tilde{f}\|_\infty \leq \varepsilon \,\}, \quad \mathbf{B}_\infty(f, \epsilon) := \bigcup_{j \in \mathbb{N}} \mathbf{B}_\infty^j(f, \epsilon). \tag{3.18}$$

Recall that (3.17) is the dual of (3.9) with $\eta = 0$, that is,

$$\begin{aligned} \sup_{\tilde{f}, \lambda} \{\, & \lambda : \ \tilde{f} - \lambda \in \mathcal{Q}_j(g);\ |f_\alpha - \tilde{f}_\alpha| \leq \varepsilon,\ \alpha \in \mathbb{N}_{2j}^n, \\ & \lambda \in \mathbb{R},\ \tilde{f} \in \mathbb{R}[\mathbf{x}]_{2j} \,\}. \end{aligned} \tag{3.19}$$

Fix $j \in \mathbb{N}$ and consider the following robust polynomial optimization problem:

$$\mathbf{P}_\varepsilon^{\max} : \quad \max_{\tilde{f} \in \mathbf{B}_\infty(f, \epsilon)} \left\{ \min_{\mathbf{x} \in \mathbf{K}} \{ \tilde{f}(\mathbf{x}) \} \right\}. \tag{3.20}$$

If in (3.20) we restrict ourselves to $\mathbf{B}_\infty^j(f, \epsilon)$ in the outer maximization problem and we replace the inner minimization by its step-j relaxation, we obtain

$$\begin{aligned} \mathbf{P}_\varepsilon^{\max,j} \ : & \max_{\tilde{f} \in \mathbf{B}_\infty^j(f, \epsilon)} \left\{ \sup_{\lambda} \{\, \lambda : \ \tilde{f} - \lambda \in \mathcal{Q}_j(g) \} \right\} \\ = & \max_{\tilde{f} \in \mathbf{B}_\infty^j(f, \epsilon)} \left\{ \inf_{\mathbf{y}} \{\, L_{\mathbf{y}}(\tilde{f}) : \ y_0 = 1,\ \mathbf{M}_j(g_\ell\, \mathbf{y}) \succeq 0,\ \ell = 0, \ldots, m \} \right\}. \end{aligned} \tag{3.21}$$

Here, we rely again on Assumption 3.1 to ensure strong duality and obtain (3.21). Problem $\mathbf{P}_\varepsilon^{\max,j}$ is a strengthening of $\mathbf{P}_\varepsilon^{\max}$ whose dual is exactly (3.17), that is, we have the following proposition.

PROPOSITION 3.6. *Under Assumption* 3.1*, solving* (3.17) *(equivalently* (3.19)*) can be interpreted as solving* exactly*, i.e., with no trace-equality tolerance, the step-j reinforcement $\mathbf{P}_\varepsilon^{\max,j}$ associated with $\mathbf{P}_\varepsilon^{\max}$.*

3.3. A two-player game interpretation. If we now assume that one can perform computations exactly, we can interpret the whole process in $\mathbf{P}_\eta^{\max,j}$ (resp., $\mathbf{P}_\varepsilon^{\max,j}$) as a two-player zero-sum game in which

- Player 1 (the solver) chooses a polynomial $\tilde{f} \in \mathbf{B}_\infty^j(f, \mathbf{K}, \eta)$ (resp., $\tilde{f} \in \mathbf{B}_\infty^j(f, \varepsilon)$);
- Player 2 (the optimizer) then selects a minimizer $\mathbf{y}^\star(\tilde{f})$ in the inner minimization of (3.21), e.g., with an exact interior point method.

As a result, Player 1 (the leader) obtains an optimal polynomial $\tilde{f}^\star \in \mathbf{B}_\infty^j(f, \mathbf{K}, \eta)$ (resp., $\tilde{f}^\star \in \mathbf{B}_\infty^j(f, \varepsilon)$) and Player 2 (the follower) obtains an associated minimizer $\mathbf{y}^\star(\tilde{f}^\star)$.

The polynomial $\tilde{f}^\star$ is the *worst* polynomial in $\mathbf{B}_\infty^j(f, \mathbf{K}, \eta)$ (resp., $\mathbf{B}_\infty^j(f, \varepsilon)$) for the step-$j$ semidefinite relaxation associated with the optimization problem

$$\min_{\mathbf{x}} \{ \tilde{f}(\mathbf{x}) : \mathbf{x} \in \mathbf{K} \} .$$

This max-min problem is then equivalent to the single min-problem (3.12) (resp., (3.17)), which is a convex relaxation whose convex criterion is not linear as it contains the sum of ℓ_∞-norm terms $\sum_{\ell=0}^m \|\mathbf{M}_{j-d_\ell}(g_\ell\, \mathbf{y})\|_*$ (resp., the ℓ_1-norm term $\|\mathbf{y}\|_1$).

Notice that in this scenario the optimizer (Player 2) is *not* active; initially he wanted to solve the convex relaxation associated with f. It is Player 1 (the adversary uncertainty in the solver) who in fact *gives* the exact algorithm his own choice of the function $\tilde{f} \in \mathbf{B}_\infty^j(f, \mathbf{K}, \eta)$ (resp., $\tilde{f} \in \mathbf{B}_\infty^j(f, \varepsilon)$). But in fact, as we are in the convex case, Theorem 2.1 implies that this max-min game is also equivalent to the min-max game. Indeed, $\mathbf{P}_\eta^{\max,j}$ is equivalent to

$$\inf_{\mathbf{y}} \max_{\tilde{f} \in \mathbf{B}_\infty^j(f,\mathbf{K},\eta)} \{ L_{\mathbf{y}}(\tilde{f}) : \, y_0 = 1, \; \mathbf{M}_j(g_\ell\, \mathbf{y}) \succeq 0, \; \ell = 0, \ldots, m \} ,$$

and $\mathbf{P}_\varepsilon^{\max,j}$ is equivalent to

$$\inf_{\mathbf{y}} \max_{\tilde{f} \in \mathbf{B}_\infty^j(f,\varepsilon)} \{ L_{\mathbf{y}}(\tilde{f}) : \, y_0 = 1, \; \mathbf{M}_j(g_\ell\, \mathbf{y}) \succeq 0, \; \ell = 0, \ldots, m \} .$$

So now in this scenario (which assumes exact computations), we have the following.

- Player 1 (the robust optimizer) chooses a feasible moment sequence $\mathbf{y}$ with $\mathbf{y}_0 = 1$ and $\mathbf{M}_{j-d_\ell}(g_\ell\, \mathbf{y}) \succeq 0$, $\ell = 0, \ldots, m$.
- When priority is given to trace equalities, Player 2 (the solver) then selects $\tilde{f}(\mathbf{y}) = \arg\max\{L_{\mathbf{y}}(\tilde{f}) : \tilde{f} \in \mathbf{B}_\infty^j(f, \mathbf{K}, \eta)\}$ to obtain the value $L_{\mathbf{y}}(f) + \eta \sum_{\ell=0}^m \|\mathbf{M}_{j-d_\ell}(g_\ell\, \mathbf{y})\|_*$.

  When priority is given to semidefiniteness inequalities, Player 2 selects $\tilde{f}(\mathbf{y}) = \arg\max\{L_{\mathbf{y}}(\tilde{f}) : \tilde{f} \in \mathbf{B}_\infty^j(f, \varepsilon)\}$ to obtain the value $L_{\mathbf{y}}(f) + \varepsilon \|\mathbf{y}\|_1$, that is, $\tilde{f}(\mathbf{y})_\alpha = f_\alpha + \mathrm{sign}(y_\alpha)\, \varepsilon$, $\alpha \in \mathbb{N}_{2j}^n$.

Here the optimizer (now Player 1) is "active" as *he* decides to compute a "robust" optimal relaxation $\mathbf{y}$ assuming uncertainty in the function f in the criterion $L_{\mathbf{y}}(f)$.

Since both scenarios are equivalent, it is fair to say that the SDP solver is indeed solving the robust convex relaxation that the optimizer would have given to a solver with exact arithmetic (if he had wanted to solve robust relaxations).

Relating to robust optimization. Suppose that there is no computation error but we want to solve a robust version of the optimization problem $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\}$ because there is some uncertainty in the coefficients of the *nominal* polynomial $f \in \mathbb{R}[\mathbf{x}]_d$. So assume that $f \in \mathbb{R}[\mathbf{x}]_d$ can be considered as potentially of degree at most $2j$ (after perturbation).

When priority is given to trace equalities, the robust optimization problem reads

$$\mathbf{P}_\eta^{\min,j}: \quad \min_{\mathbf{x}\in\mathbf{K}} \Big\{ \max_{\tilde f \in \mathbf{B}_\infty^j(f,\mathbf{K},\eta)} \{\tilde f(\mathbf{x})\} \Big\}. \tag{3.22}$$

A straightforward calculation reduces (3.22) to

$$\mathbf{P}_\eta^{\min,j}: \quad \min_{\mathbf{x}\in\mathbf{K}} \left[f(\mathbf{x}) + \eta \sum_{\beta\in\mathbb{N}^n_{j-d_\ell}} \mathbf{x}^{2\beta}\, g_\ell(\mathbf{x}) \right], \tag{3.23}$$

which is a polynomial optimization problem.

THEOREM 3.7. *Suppose that Assumption* 3.1 *holds. Assume that after solving SDP* (3.12)*, one obtains* $\mathbf{y}^\star$ *such that* $\mathbf{M}_j(\mathbf{y}^\star)$ *is a rank-one matrix. Then, the optimal value of* $\mathbf{P}_\eta^{\min,j}$ *is equal to* ρ_η^j *and* $\mathbf{P}_\eta^{\min,j}$ *is equivalent to* $\mathbf{P}_\eta^{\max,j}$.

Proof. Since $\mathbf{M}_j(\mathbf{y}^\star)$ is a rank-one matrix, the sequence $\mathbf{y}^\star$ comes from a Dirac measure supported on $\mathbf{x}^\star \in \mathbf{K}$. Then one has

$$L_{\mathbf{y}^\star}(f) + \eta \sum_{\ell=0}^m \|\mathbf{M}_{j-d_\ell}(g_\ell\, \mathbf{y}^\star)\|_* = f(\mathbf{x}^\star) + \eta \sum_{\beta\in\mathbb{N}^n_{j-d_\ell}} \mathbf{x}^{\star 2\beta}\, g_\ell(\mathbf{x}^\star)\,,$$

Let $\mathcal{P}(\mathbf{K})$ be the space of probability measures supported on $\mathbf{K}$. Then one has

$$\begin{aligned} f(\mathbf{x}^\star) + \eta \sum_{\ell=0}^m \sum_{\beta\in\mathbb{N}^n_{j-d_\ell}} \mathbf{x}^{\star 2\beta}\, g_\ell(\mathbf{x}^\star) &\geq \min_{\mathbf{x}\in\mathbf{K}} \left[f(\mathbf{x}) + \eta \sum_{\ell=0}^m \sum_{\beta\in\mathbb{N}^n_{j-d_\ell}} \mathbf{x}^{2\beta}\, g_\ell(\mathbf{x}) \right] \\ &= \inf_{\mu\in\mathcal{P}(\mathbf{K})} \left[\int f d\mu + \eta \sum_{\ell=0}^m \sum_{\beta\in\mathbb{N}^n_{j-d_\ell}} \int \mathbf{x}^{2\beta}\, g_\ell(\mathbf{x}) d\mu \right] \\ &\geq \rho_\eta^j = L_{\mathbf{y}^\star}(f) + \eta \sum_{\ell=0}^m \|\mathbf{M}_{j-d_\ell}(g_\ell\, \mathbf{y}^\star)\|_*\,. \end{aligned}$$

This implies that $\mathbf{x}^\star$ is the unique optimal solution of $\mathbf{P}_\eta^{\min,j}$ and that the optimal value of $\mathbf{P}_\eta^{\min,j}$ is equal to ρ_η^j. Eventually, Proposition 3.4 yields the desired equivalence. ◻

When priority is given to semidefiniteness inequalities, the robust optimization problem reads

$$\mathbf{P}_\varepsilon^{\min,j}: \quad \min_{\mathbf{x}\in\mathbf{K}} \Big\{ \max_{\tilde f \in \mathbf{B}_\infty^j(f,\varepsilon)} \{\tilde f(\mathbf{x})\} \Big\}. \tag{3.24}$$

It is easy to see that (3.24) reduces to

$$\mathbf{P}_\varepsilon^{\min,j}: \quad \min_{\mathbf{x}\in\mathbf{K}} \left[f(\mathbf{x}) + \varepsilon \sum_{\alpha\in\mathbb{N}^n_{2j}} |\mathbf{x}^\alpha| \right], \tag{3.25}$$

which is *not* a polynomial optimization problem (but is still a semialgebraic optimization problem). As for Theorem 3.7, we can prove the following result.

THEOREM 3.8. *Suppose that Assumption* 3.1 *holds. Assume that after solving SDP* (3.17)*, one obtains* $\mathbf{y}^\star$ *such that* $\mathbf{M}_j(\mathbf{y}^\star)$ *is a rank-one matrix. Then, the optimal value of* $\mathbf{P}_\varepsilon^{\min,j}$ *is equal to* ρ_ε^j *and* $\mathbf{P}_\varepsilon^{\min,j}$ *is equivalent to* $\mathbf{P}_\varepsilon^{\max,j}$.

Notice an important conceptual difference between the two approaches. In the latter one, i.e., when considering $\mathbf{P}_\eta^{\min}$ (resp., $\mathbf{P}_\varepsilon^{\min}$), the user is active. Indeed the user decides to choose some optimal $\hat{f} \in \mathbf{B}_\infty^j(f, \mathbf{K}, \eta)$ (resp., $\mathbf{B}_\infty^j(f, \varepsilon)$). In the former one, i.e., when considering $\mathbf{P}_\eta^{\max}$ (resp., $\mathbf{P}_\varepsilon^{\max}$), the user is passive, as indeed he imposes f but the solver decides to choose some optimal $f^\star \in \mathbf{B}_\infty^j(f, \mathbf{K}, \eta)$ (resp., $\mathbf{B}_\infty^j(f, \varepsilon)$).

If after solving SDP (3.12) (resp., SDP (3.17)), one obtains $\mathbf{y}^\star$ where $\mathbf{M}_j(\mathbf{y}^\star)$ is rank-one (which is to be expected), one obtains the same solution. In other words, we can interpret what the solver does as performing robust polynomial optimization.

In what follows, we show how this interpretation relates to a more general robust SDP framework, when priority is given to semidefiniteness inequalities.

3.4. Link with robust semidefinite programming. Let $\mathbf{c} = (c_j) \in \mathbb{R}^n$, $\mathbf{F}_j$ be a real symmetric $t \times t$ matrix, $j = 0, 1, \ldots, n$, and let $\mathbf{F}(\mathbf{y}) := \sum_{j=1}^n \mathbf{F}_j\, y_j - \mathbf{F}_0$. Consider the canonical semidefinite program

$$\mathbf{P}: \quad \inf_{\mathbf{y}} \{\, \mathbf{c}^T\mathbf{y} : \mathbf{F}(\mathbf{y}) \succeq 0 \,\} \tag{3.26}$$

with dual

$$\mathbf{P}^*: \quad \sup_{\mathbf{X} \succeq 0} \{ \langle \mathbf{F}_0, \mathbf{X} \rangle : \langle \mathbf{F}_j, \mathbf{X} \rangle = c_j,\ j = 1, \ldots, n \,\}. \tag{3.27}$$

Given $\varepsilon > 0$ fixed, let $\mathbf{B}_\infty(\mathbf{c}, \epsilon) := \{\, \tilde{\mathbf{c}} : \|\tilde{\mathbf{c}} - \mathbf{c}\|_\infty \le \varepsilon \,\}$ and consider the max-min problem associated with $\mathbf{P}$:

$$\rho = \max_{\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)} \inf_{\mathbf{y}} \{\, \tilde{\mathbf{c}}^T\, \mathbf{y} : \mathbf{F}(\mathbf{y}) \succeq 0 \,\}. \tag{3.28}$$

As in section 3.3, there is a simple two-player game interpretation of (3.28). Player 1 (the leader) searches for the "best" cost function $\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)$, which is "robust" against the *worst* decision $\mathbf{y}$ made by Player 2 (the follower, the decision maker), once Player 1's choice $\tilde{\mathbf{c}}$ is known.

PROPOSITION 3.9. *Assume that there exists* $\hat{\mathbf{y}}$ *such that* $\mathbf{F}(\hat{\mathbf{y}}) \succ 0$. *Then solving the max-min problem* (3.28) *is equivalent to solving*

$$\inf_{\mathbf{y}} \{\, \mathbf{c}^T\, \mathbf{y} + \varepsilon\, \|\mathbf{y}\|_1 : \mathbf{F}(\mathbf{y}) \succeq 0 \,\}. \tag{3.29}$$

Proof. $\mathbf{F}(\hat{\mathbf{y}}) \succ 0$ implies that Slater's condition holds for the inner (minimization) SDP of (3.28). Therefore, by standard conic duality,

$$\rho = \max_{\tilde{\mathbf{c}}} \sup_{\mathbf{X} \succeq 0} \{\, \langle \mathbf{F}_0, \mathbf{X} \rangle : \langle \mathbf{F}_j, \mathbf{X} \rangle = \tilde{c}_j\,,\ |\, \tilde{c}_j - c_j \,| \le \varepsilon,\ j = 1, \ldots, n \,\}, \tag{3.30}$$

which in turn is equivalent to

$$\sup_{\mathbf{X} \succeq 0} \{\, \langle \mathbf{F}_0, \mathbf{X} \rangle : |\langle \mathbf{F}_j, \mathbf{X} \rangle - c_j| \le \varepsilon\,,\ j = 1, \ldots, n \,\}. \tag{3.31}$$

As in the proof of Proposition 3.2, we prove that the dual of SDP (3.31) is (3.29). ∎

So again, with an appropriate value of ε related the numerical precision of SDP solvers, (3.31) can be considered as a fair model for treating inaccuracies by relaxing the equality constraints of (3.27) up to some tolerance level ε. That is, instead of solving exactly (3.27) with nominal criterion $\mathbf{c}$, Player 1 (the SDP solver) is considering a related robust version where it solves (exactly) (3.27) but now with some optimal choice of a new cost vector $\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)$. But this is a robustness point of view from the solver (*not* from the decision maker) and the resulting robust solution is some optimal cost vector $\tilde{\mathbf{c}}^* \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)$.

In the particular case of SDP relaxations for polynomial optimization, we retrieve (3.17) as an instance of (3.29) and (3.19) as an instance of (3.31).

Robust SDP. On the other hand, the objective function $\tilde{\mathbf{c}}^T\mathbf{y}$ is bilinear in $(\tilde{\mathbf{c}}, \mathbf{y})$, the set $\mathbf{B}^j_\infty(\mathbf{c}, \varepsilon)$ is convex and compact, and the set $\mathbf{Y} := \{\mathbf{y} : \mathbf{F}(\mathbf{y}) \succeq 0\}$ is convex. Hence, by Theorem 2.1, (3.28) is equivalent to solving the min-max problem

$$\rho = \inf_{\mathbf{y}} \left\{ \max_{\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c},\varepsilon)} \{ \tilde{\mathbf{c}}^T\mathbf{y} \} : \mathbf{F}(\mathbf{y}) \succeq 0 \right\}, \tag{3.32}$$

which is a "robust" version of (3.26) from the point of view of the decision maker when there is uncertainty in the cost vector. That is, the cost vector $\tilde{\mathbf{c}}$ is not known exactly and belongs to the uncertainty set $\mathbf{B}_\infty(\mathbf{c}, \varepsilon)$. The decision maker has to make a robust decision $\mathbf{y}^*$ which is the best against all possible values of the cost function $\tilde{\mathbf{c}} \in \mathbf{B}_\infty(\mathbf{c}, \varepsilon)$. This well-known latter point of view is that of *robust optimization* in the presence of uncertainty for the cost vector; see, e.g., [4].

So if the latter robustness point of view (of the decision maker) is well known, what is perhaps less well known (but not so surprising) is that it can be interpreted in terms of a robustness point of view from an inexact "solver" when treating equality constraints with inaccuracies in a problem with nominal criterion. Given problem (3.26) with nominal criterion $\mathbf{c}$, and without being asked to do so, the solver behaves *as if* it is *exactly* solving the robust version (3.32) (from the decision-maker viewpoint), whereas the decision maker is willing to solve (3.26) exactly. In other words, Sion's minimax theorem validates the informal (and not surprising) statement that the treatment of inaccuracies by the SDP solver can be viewed as a robust treatment of uncertainties in the cost vector.

However, in the case of SDP relaxations for polynomial optimization, this behavior is indeed more surprising and even spectacular. Indeed, some unconstrained optimization instances, such as minimizing Motzkin-like polynomials (i.e., when $f - f^*$ is not SOS), cannot be theoretically handled by SDP relaxations (assuming that one relies on exact SDP solvers). Yet, double floating point SDP solvers solve them in a practical manner, provided that higher-order relaxations are allowed so that a polynomial of degree d can be (and indeed is!) treated as a higher degree polynomial (but with zero coefficients for monomials of degree higher than d).

In general, similar phenomena can occur while relying on general floating point algorithms. We presume that they could also appear when handling polynomial optimization problems with alternative convex programming relaxations relying on interior-point algorithms, for instance linear/geometric programming.

4. Examples.

All experimental results are obtained by computing the solutions of the primal-dual SDP relaxations (3.4) and (3.5) of problem $\mathbf{P}$. These SDP relaxations are implemented in the `RealCertify` [12] library, available within MAPLE, and interfaced with the SDP solvers SDPA [22] and SDPA-GMP [13].

For the two upcoming examples, we rely on the procedure described in [5] to extract the approximate global minimizer(s) of some given objective polynomial functions. We compare the results obtained with (1) the SDPA solver implemented in double floating point precision, which corresponds to $\epsilon = 10^{-7}$, and (2) the arbitrary-precision SDPA-GMP solver, with $\epsilon = 10^{-30}$. The value of our robust-noise model parameter ε roughly matches with that of the parameter `epsilonStar` of SDPA.

We also noticed that decreasing the value of the SDPA parameter `lambdaStar` seems to boil down to increasing the value of our robust-noise model parameter η. An expected justification is that `lambdaStar` is used to determine a starting point $\mathbf{X}^0$ for the interior-point method, i.e., it is such that $\mathbf{X}^0 = \texttt{lambdaStar} \times \mathbf{I}$ (the default value of `lambdaStar` is equal to 10^2 in SDPA and is equal to 10^4 in SDPA-GMP). A similar behavior occurs when decreasing the value of the parameter `betaBar`, which controls the search direction of the interior-point method when the matrix $\mathbf{X}$ is not positive semidefinite.

However, the correlation between the values of `lambdaStar` (and `betaBar`) and η appears to be nontrivial. Thus, our robust-noise model would be theoretically valid if one could impose the value of a parameter η, ensuring that $\mathbf{X} \succeq -\eta\mathbf{I}$ when the interior-point method terminates. To the best of our knowledge, this feature happens to be unavailable in modern SDP solvers. For that reason, our experimental comparisons are performed by changing the value of `epsilonStar` in the parameter file of the SDP solver.

4.1. Motzkin polynomial. Here, we consider the Motkzin polynomial $f = \frac{1}{27} + x_1^2 x_2^2 (x_1^2 + x_2^2 - 1)$. This polynomial is nonnegative but is not SOS. The minimum $f^\star$ of f is 0 and f has four global minimizers with coordinates $x_1 = \pm\frac{\sqrt{3}}{3}$ and $x_2 = \pm\frac{\sqrt{3}}{3}$. As noted in [5, section 4], one can retrieve these global minimizers by solving the primal-dual SDP relaxations (3.4) and (3.5) of problem $\mathbf{P}$ at relaxation order $j = 8$.

(1) With $\epsilon = 10^{-7}$, we obtain an approximate lower bound of $-1.81 \cdot 10^{-4} \leq f^\star$ as well as the four global minimizers of f with the extraction procedure. The dual SDP (3.5) allows us to retrieve the approximate SOS decomposition $f(\mathbf{x}) = \sigma(\mathbf{x}) + r(\mathbf{x})$, where σ is an SOS polynomial and the corresponding polynomial remainder r has coefficients of approximately equal magnitude, which is less than 10^{-8}.

(2) With $\epsilon = 10^{-30}$, we obtain an approximate lower bound of $-1.83 \cdot 10^{1} \leq f^\star$ and the extraction procedure fails. The corresponding polynomial remainder has coefficients of magnitude less than 10^{-31}.

We notice that the support of r contains only terms of even degrees, i.e., terms of the form $\mathbf{x}^{2\beta}$ with $|\beta| \leq 8$. Hence we consider a perturbation $\tilde{f}_\gamma$ of f defined by

$$\tilde{f}_\gamma(\mathbf{x}) = f(\mathbf{x}) + \gamma \sum_{|\beta| \leq j} \mathbf{x}^{2\beta}$$

with $\gamma = 10^{-8}$. By solving the SDP relaxation (with $j = 8$) associated with $\tilde{f}_\gamma$, with $\varepsilon = 10^{-30}$, we again retrieve the four global minimizers of f.

4.2. Univariate polynomial with minimizers of different magnitudes. We start by considering the univariate optimization problem

$$f^\star = \min_{x \in \mathbb{R}} f(x)$$

with $f(x) = (x - 100)^2((x-1)^2 + \frac{\gamma}{99^2})$ and $\gamma \geq 0$.

Note that the minimum of f is $f^\star = 0 = f(100)$ and $f(1) = \gamma$.

We first examine the case where $\gamma = 0$. In this case, f has two global minimizers 1 and 100. At relaxation order j, with $2 \leq j \leq 5$, we retrieve the following results (rounded to four significant digits).

(1) With $\epsilon = 10^{-7}$, we obtain $\hat{x}^{(1)} = 0.9999 \simeq 1$, corresponding to the smallest global minimizer of f.

(2) With $\epsilon = 10^{-30}$, we obtain $\hat{x} = 50.5000 = \frac{1+100}{2}$, corresponding to the average of the two global minimizers of f.

We also used the `realroot` procedure, available within MAPLE, to compute the local minimizers of the following function on $[0, \infty)$:

$$\tilde{f}_{\varepsilon,j}(x) = f(x) + \varepsilon \sum_{|\alpha| \leq 2j} |x^\alpha| = f(x) + \varepsilon \sum_{|\alpha| \leq 2j} x^\alpha \,. \tag{4.1}$$

(1) With $\epsilon = 10^{-7}$, we obtain $\tilde{x}^{(1)} = 0.9961 \simeq \hat{x}^{(1)}$.

(2) With $\epsilon = 10^{-30}$, we obtain $\tilde{x}^{(1)} = 0.9961 \simeq \hat{x}^{(1)}$ and $\tilde{x}^{(2)} = 99.9960 \simeq 100$, the largest global minimizer of f. The corresponding values of $\tilde{f}_{\varepsilon,j}$ are 0.1496 and 0.1495, respectively.

These experiments confirm our explanation that the solver computes the solution of SDP relaxations associated with the perturbed function $\tilde{f}_{\varepsilon,j}$ from (4.1). With the double floating point precision (1), this perturbed function has a single minimizer, retrieved by the extraction procedure. With the higher precision (2), this perturbed function has two local minimizers, whose average is retrieved by the extraction procedure.

Next, we examine the case where $\gamma = 10^{-3}$. In this case, f has a single global minimizer, equal to 100 and another local minimizer. At relaxation order j, with $2 \leq j \leq 5$, we retrieve the following results (rounded to four significant digits).

(1) With $\epsilon = 10^{-7}$, we obtain $\hat{x}^{(1)} = 0.9999 \simeq 1$, corresponding to the smallest global minimizer of f when $\gamma = 0$.

(2) With $\epsilon = 10^{-30}$, we obtain $\hat{x}^{(2)} = 99.1593 \simeq 100$, corresponding to the single global minimizer of f.

We also compute the local minimizers of $\tilde{f}_{\varepsilon,j}$ with `realroot`.

(1) With $\epsilon = 10^{-7}$, we obtain $\tilde{x}^{(1)} = 1.0039 \simeq \hat{x}^{(1)}$.

(2) With $\epsilon = 10^{-30}$, we obtain $\tilde{x}^{(1)} = 1.0039 \simeq \hat{x}^{(1)}$ and $\tilde{x}^{(2)} = 99.9961 \simeq 100$, the single global minimizer of f. The corresponding values of $\tilde{f}_{\varepsilon,j}$ are 0.1505 and 0.1495, respectively. This confirms that $\tilde{x}^{(2)}$ is the single global minimizer of $\tilde{f}_{\varepsilon,j}$, approximately extracted as $\hat{x}^{(2)}$.

Here again, our robust-noise model, relying on the perturbed polynomial function $\tilde{f}_{\varepsilon,j}$, fits with the above experimental observations. This perturbed function has a single global minimizer whose value depends on the parameter ε, and which can be approximately retrieved by the extraction procedure.

5. Discussion. By considering the hierarchy of SDP relaxations associated with a given polynomial optimization problem, we are faced with a dilemma when relying on numerical SDP solvers. On the one hand, we might want to increase the precision of the solver to get rid of the numerical uncertainties and obtain an accurate solution of the SDP relaxations. On the other hand, working with low precision may allow us to obtain hints related to the solution of the initial problem. This has already happened in both commutative and noncommutative contexts when computing the global minimizers of the Motzkin polynomial in [5] or the bosonic energy levels from [14].

Our theoretical robust-noise model could be extended to problems addressed with structured SDP programs (as, for instance, the moment and localizing matrices coming from polynomial optimization problems). We believe that the use of "inaccurate" SDP solvers could also provide hints for the solutions of such problems. One could estimate how close the optimal values of duals (3.12) and (3.17) of noise models are to the optimal values of the initial optimization problem **P**. For some instances in this article and other papers from the literature, the optimal values seem to not exceed the optimal value of **P** for higher orders sufficiently large. Such experimental observations remain to be explained and/or validated.

REFERENCES

[1] E. Andersen and K. Andersen, *The Mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm*, in High Performance Optimization, H. Frenk, K. Roos, T. Terlaky, and S. Zhang, eds., Appl. Optim. 33, Springer, Boston, MA, 2000, pp. 197–232.

[2] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton Ser. Appl. Math., Princeton University Press, Princeton, NJ, 2009.

[3] S. Burgdorf, I. Klep, and J. Povh, *Optimization of Polynomials in Non-commuting Variables*, SpringerBriefs Math., Springer, Cham, 2016.

[4] L. E. Ghaoui, F. Oustry, and H. Lebert, *Robust solutions to uncertain semidefinite programs*, SIAM J. Optim., 9 (1998), pp. 33–52.

[5] D. Henrion and J.-B. Lasserre, *Detecting global optimality and extracting solutions in GloptiPoly*, in Positive Polynomials in Control, Lect. Notes Control Inf. Sci. 312, Springer, Berlin, Heidelberg, 2005, pp. 293–310.

[6] D. Henrion, S. Naldi, and M. Safey El Din, *Exact algorithms for linear matrix inequalities*, SIAM J. Optim., 26 (2016), pp. 2512–2539.

[7] C. Josz and D. Henrion, *Strong duality in Lasserre's hierarchy for polynomial optimization*, Optim. Lett., 10 (2016), pp. 3–10.

[8] J.-B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.

[9] J.-B. Lasserre, *A sum of squares approximation of nonnegative polynomials*, SIAM Rev., 49 (2007), pp. 651–669.

[10] J.-B. Lasserre, *Moments, Positive Polynomials and Their Applications*, Imperial College Press Optim. Ser., Imperial College Press, London, 2009.

[11] V. Magron and M. S. E. Din, *On exact Polya and Putinar's representations*, in Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation (ISSAC'18), ACM Press, New York, 2018, pp. 279–286.

[12] V. Magron and M. S. E. Din, *RealCertify: A Maple Package for Certifying Non-negativity*, preprint, http://arxiv.org/abs/1805.02201, 2018.

[13] M. Nakata, *A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: SDPA-GMP, -QD, and -DD*, in Proceedings of the 2010 IEEE International Symposium on Computer-Aided Control System Design, IEEE Press, Piscataway, NJ, 2010, pp. 29–34.

[14] M. Navascués, A. García-Sáez, A. Acín, S. Pironio, and M. B. Plenio, *A paradox in bosonic energy computations via semidefinite programming relaxations*, New J. Phys., 15 (2013), 023026.

[15] J. Nie, *Certifying convergence of Lasserre's hierarchy via flat truncation*, Math. Program., 142 (2013), pp. 485–510.

[16] J. Nie, *Optimality conditions and finite convergence of Lasserre's hierarchy*, Math. Program., 146 (2014), pp. 97–121.

[17] G. Pataki, *Bad semidefinite programs: They all look the same*, SIAM J. Optim., 27 (2017), pp. 146–172, https://doi.org/10.1137/15M1041924.

[18] M. Sion, *On general minimax theorems*, Pacific J. Math., 8 (1958), pp. 171–176.

[19] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, J. Optim. Methods Softw., 11 (1999), pp. 625–653.

[20] H. Waki, *How to generate weakly infeasible semidefinite programs via Lasserre's relaxations for polynomial optimization*, Optim. Lett., 6 (2012), pp. 1883–1896, https://doi.org/10.1007/s11590-011-0384-1.

[21] H. Waki, M. Nakata, and M. Muramatsu, *Strange behaviors of interior-point methods for solving semidefinite programming problems in polynomial optimization*, Comput. Optim. Appl., 53 (2012), pp. 823–844.

[22] M. Yamashita, K. Fujisawa, K. Nakata, M. Nakata, M. Fukuda, K. Kobayashi, and K. Goto, *A High-Performance Software Package for Semidefinite Programs: SDPA7*, Technical report, Department of Information Sciences, Tokyo Institute of Technology, 2010; available at http://www.optimization-online.org/DB_FILE/2010/01/2531.pdf.