THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse III - Paul Sabatier*
Discipline ou spécialité : *Mathématiques*

Présentée et soutenue par

PHAN THANH TUNG

Le *04 octobre 2012*

Titre : *Contributions to the Moment-SOS Approach in Global Polynomial Optimization*

JURY

Jean-Baptiste HIRIART-URRUTY, Université Paul Sabatier
Monique LAURENT, Tilburg University
Jean Bernard LASSERRE, LAAS-CNRS
Leo LIBERTI, l'Ecole Polytechnique
Dinh Tao PHAM, l'INSA de Rouen

Ecole doctorale : *Mathématiques Informatique Télécommunications de Toulouse (MITT)*

Unité de recherche : *LAAS - CNRS*

Directeur(s) de Thèse : *Jean Bernard LASSERRE*

“Whatever you do, work at it with all your heart..."

Acknowledgements

Foremost, I would like to express my deep and sincere gratitude to my supervisor, Professor Jean Bernard Lasserre for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. Without his encouraging and personal guidance from the very first days in Toulouse until the end of three years, this dissertation would not have been possible.

Besides my advisor, I would also like to thank all the members of group MAC for providing me the financial means and making it a convivial place to work. In particular, I would like to thank Professor Didier Henrion who was always accessible and willing to help with his insightful comments for my work.

I would also like to convey thanks to the French Government, Vietnam Ministry of Education and Training, Vietnam Commerce University for supporting me the fellowship and perfect living conditions during the time I studied in France.

My deepest gratitude goes to my beloved families, my parents Phan Trong Ngo and Nguyen Thu Tam, my sister Phan Thuy Thuy and my girlfriend Dinh Thi Huyen Trang. Thank you for being ever understanding and endless love. Your prayers and encouragement have been helping me throughout the duration of my studies.

Finally, my thanks go to all the Vietnamese friends in Toulouse, especially to professors and students of Mathematics in University Paul Sabatier, professor Nguyen Tien Dung, friends Minh, Lien, Minh blanc, Binh, Yen, Hung, Ngoc...and not forgetting little Paul. Thank you very much.

Tung Phan Thanh, Toulouse 2012.

Contents

List of Tables

Chapter 1

General introduction

The general context

Polynomial Optimization is concerned with optimization problems of the form

$$\begin{aligned} \mathbf{P}: \quad f^* &= \inf_{\mathbf{x}} \{ f(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K} \} \\ \text{with } \mathbf{K} &= \{ \mathbf{x} \in \mathbb{R}^n \,:\, g_j(\mathbf{x}) \,\geq\, 0, \quad j = 1, \ldots, m \}, \end{aligned} \tag{1.1}$$

where f, g_j, $j = 1, \ldots, m$, are polynomials. Why then give the name “Polynomial Optimization" since $\mathbf{P}$ in (1.1) is just a particular case of the general *Nonlinear Programming Problem* (NLP)?

The reason why this name has emerged in the last decade is because thanks to powerful results of Real Algebraic Geometry it has been possible to define a systematic numerical scheme to compute, or at least approximate as closely as desired, the **global** minimum f^*. In a more general framework where f and g_j are not polynomials, no such systematic procedure exists. Of course, in the case where the g_j's define relatively simple feasible sets (e.g., a box) some brute force discretization procedure is possible. Also for Mixed-Integer Linear and convex Mixed-Integer Non Linear Problems (MILPs and convex MINLPs), a variety of effective exact solution methods have been developed in the last decades. For examples, Branch & Bound schemes [24], outer approximation [16], LP/NLP-based Branch & Bound [63], the extended cutting-plane method [78], to cite a few. We refer the interested reader to the recent contributions [48, 69, 79] and the references therein for a deep analysis of the theory and of some effective software packages that came from theoretical developments (e.g., CPLEX [32], XPRESS-MP [12], BONMIN [9], DICOPT [36] and LaGO [60]). On the other hand, in non-convex MINLP and NLP in general, the situation is more contrasted. Even though several software packages can sometimes provide exact or good approximate solutions (e.g. BARON [66], α**BB** [2], Couenne [7], LINDO-Global [51]), the computational effort is in general more important because the problems are more difficult. Therefore one is often satisfied with finding only a local minimizer $\mathbf{x}^* \in \mathbf{K}$ with $f(\mathbf{x}^*) > f^*$. For further details the interested reader is referred to the excellent recent surveys [10, 27, 48].

In NLP the main tools are from Real and Convex Analysis as well as Linear Algebra, whereas in Polynomial Optimization additional results from Real Algebraic Geometry can also be exploited to derive powerful global representation theorems for polynomial that are positive on certain sets. This is why polynomial optimization is a subfield of NLP that deserves special attention. Examples of such represention theorems are the Positivstellensatze[1] of Handelman [25], Krivine [39], Stengle [71], Schmüdgen [68] and Putinar [62] (the latter being especially useful for the moment-SOS approach described in this thesis).

In particular, a popular approach to compute (or at least approximate) f^* is via solving a hierarchy of convex relaxations whose associated sequence of optimal values is monotone non decreasing and converges to f^*. Convergence of this numerical scheme is proved by invoking these powerful representation theorems. Depending on which type of representation is used, the resulting convex relaxations in the hierarchy are either linear programs (LP) or semidefinite programs (SDP). For more details on this methodology the interested reader is referred to [44] , [47] and [61]

It turns out that the latter hierarchy (which we call the *moment-SOS* approach) is more efficient and preferable. On the other hand, even though this approach is very efficient, the size the semidefinite relaxations grows rapidly with the rank in the hierarchy and becomes prohibitive. Typically, if problem $\mathbf{P}$ in (1.1) has n variables then the d-th semidefinite relaxation in the hierarchy has $O(n^{2d})$ variables and semidefiniteness constraints of matrices with size $O(n^d)$. Therefore, in view of the present status of public semidefinite solvers available, so far this approach is limited to small to medium size problems $\mathbf{P}$ only. However, as is often the case in large scale problems, if symmetries and/or some structured sparsity are present in problem $\mathbf{P}$, they can be exploited right from the beginning to define appropriate hierarchies of semidefinite relaxations of much smaller size. For more details on how to use symmetries the interested reader is referred to e.g. [19, 47, 74] and for structured sparsity to e.g. [37, 76, 77].

[1] In german, “Positivstellensatz" is a theorem on positivity.

In this thesis we are interested in problems $\mathbf{P}$ in (1.1) where symmetries and/or structured sparsity are not easy to detect or to exploit, and where only a few (or even no) semidefinite relaxations of the moment-SOS approach can be implemented. And the issue we investigate is:

How can the moment-SOS methodology be still used to help solve such problems $\mathbf{P}$*?*

Our contribution

We provide two applications of the moment-SOS approach to help solve $\mathbf{P}$ in two different contexts. Namely:

• In a first contribution we consider MINLP problems on a box $\mathbf{B} = [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n$ and propose a moment-SOS approach to construct *polynomial convex underestimators* for the objective function f (if non convex) and for $-g_j$ if in the constraint $g_j(\mathbf{x}) \geq 0$ the polynomial g_j is not concave. Convex underestimators are used in most MINLP algorithms to compute efficiently lower bounds at each node of a search tree in an associated Branch & Bound scheme to approximate f^*. The interested reader is referred to e.g. [10, 27, 48, 69, 79] for various contributions to solve MINLP problems and to [4, 11, 57, 22, 49, 30, 67, 73] for typical approaches to obtain convex underestimators.

In most applications of interest the size of the resulting MINLPs is large but the polynomial f is a sum $\sum_\ell f_\ell$ polynomials f_ℓ with a few variables only; for instance think of f being of low degree d, hence a sum of monomials f_ℓ, each f_ℓ having no more than d variables. Similarly for nonlinear constraints, in many cases each nonlinear polynomial g_j is concerned with very few variables only and the coupling of all (many) variables $\mathbf{x}$ is through the coupling of constraints and the criterion f. Therefore for practical efficiency, a popular convex under estimator of f is the sum $\sum_\ell \tilde{f}_\ell$ where each $\tilde{f}_\ell$ is a convex underestimator of f_ℓ.

Hence we work in the context where one wishes to find a convex underestimator of a non convex polynomial f of a few variables on a box $\mathbf{B}$. The novelty with previous works on this topic is that we want to compute a polynomial convex underestimator p_d of f that minimizes the important *tightness* criterion $\int_{\mathbf{B}} |f - h|\, d\mathbf{x}$ over all convex polynomials h of degree d fixed. Indeed in previous works for computing a convex underestimator $\mathcal{L}$ of f, this tightness criterion is not taken into account *directly*. For instance, the popular α**BB** underestimator is of the form $\mathcal{L} = f - \sum_i \alpha_i(x_i^U - x_i)(x_i - x_i^L)$ and various heuristics are proposed to compute appropriate coefficients α_i's to ensure that $\mathcal{L}$ is convex on $\mathbf{B}$; see e.g. [1, 2, 3, 4, 6, 22, 23, 57]. It turns out that the moment-SOS approach is well suited to compute a polynomial convex underestimator p_d that minimizes the tightness criterion and numerical experiments on a sample of non trivial examples show that p_d outperforms $\mathcal{L}$ not only with respect to the tightness score but also in terms of the resulting lower bounds obtained by minimizing respectively p_d and $\mathcal{L}$ on $\mathbf{B}$. Similar improvements also occur when we use the moment-SOS underestimator instead of the α**BB**-one in refinements of the α**BB** method proposed in e.g. [22, 23] and [57].

• In a second contribution we consider problems $\mathbf{P}$ for which only a few semidefinite relaxations of the moment-SOS approach can be implemented, e.g., the first in the hierarchy. So such problems are not *very* large as one should be able to implement at least the first semidefinite relaxation in the moment-SOS hierarchy. However if one uses specialized softwares as in e.g. [65, 26] the first semidefinite relaxation can be solved approximately for problems $\mathbf{P}$ of significant size (say $n > 200$). Also, if some structured sparsity is present in the problem data, appropriate sparse versions of the standard semidefinite relaxations can be defined and problems of up to say a thousand variables can be handled. In general solving this semidefinite relaxation provides only a lower bound on f^* and so the question is: How can we use an optimal solution of this relaxation to help find a feasible solution of $\mathbf{P}$?

Such an approach has already been proposed for some $0/1$ programs. In particular for the MAXCUT problem, the celebrated Goemans and Williamson's randomized rounding algorithm uses an optimal solution of a semidefinite relaxation (in fact the first in the moment-SOS hierarchy to approximate f^*) to construct a feasible solution $\mathbf{x} \in \{-1, 1\}^n$ with a performance guarantee; see [20]. This technique has been extended with success to some other combinatorial optimizations problems; for more details on this issue the interested reader is referred to e.g. [**?**, 14, 31, 33, 46] and the many references therein.

Our contribution is to propose an algorithm that also uses an optimal solution of a semidefinite relaxation in the moment-SOS hierarchy (in fact a slight modification) to provide a feasible solution for the initial optimization problem but with no rounding procedure. The rationale behind the algorithm is borrowed from results in [45] for parametric polynomial optimization. In the present context, we treat the first variable x_1 of $\mathbf{x} = (x_1, \ldots, x_n)$ as a *parameter* in some bounded interval

$\mathbf{Y} \subset \mathbb{R}$. Notice that $f^* = \min\{J(y) : y \in \mathbf{Y}\}$ where J is the optimal value function $y \mapsto J(y) := \inf\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}; x_1 = y\}$. That is one has reduced the original n-dimensional optimization problem $\mathbf{P}$ to an equivalent one-dimensional optimization problem on a interval. But of course determining the optimal value function J is even more complicated than $\mathbf{P}$ as one has to determine a function (instead of a point in $\mathbb{R}^n$), an infinite-dimensional problem. But the idea is to approximate J on $\mathbf{Y}$ by a univariate polynomial $y \mapsto p_d(y)$ of degree d and fortunately, computing such a univariate polynomial is possible via solving a semidefinite relaxation associated with the parameter optimization problem. The degree d of p_d is related to the size of this semidefinite relaxation. The higher the degree d is, the better is the approximation of $J(y)$ by $p_d(y)$ and in fact one may show that $p_d \to J$ in a strong sense on $\mathbf{Y}$.

But of course the resulting semidefinite relaxation becomes harder (or impossible) to solve as d increases and so in practice d is fixed to a small value. Once the univariate polynomial p_d has been determined, one computes $\tilde{x}_1 \in \mathbf{Y}$ that minimizes p_d on $\mathbf{Y}$, a convex optimization problem that can be solved efficiently. The process is iterated to compute $\tilde{x}_2$ in a similar manner, and so on, until a point $\tilde{\mathbf{x}} \in \mathbb{R}^n$ has been computed. Finally, as $\tilde{\mathbf{x}}$ is not feasible in general, we then use $\tilde{\mathbf{x}}$ as a starting point for a local optimization procedure to find a final feasible point $\mathbf{x} \in \mathbf{K}$. When $\mathbf{K}$ is convex, the following variant is implemented. After having computed $\tilde{x}_1$ as indicated, $\tilde{x}_2$ is computed with x_1 fixed at the value $\tilde{x}_1$, and $\tilde{x}_3$ is computed with x_1 and x_2 fixed at the values $\tilde{x}_1$ and $\tilde{x}_2$ respectively, etc., so that the resulting point $\tilde{\mathbf{x}}$ is feasible, i.e., $\tilde{\mathbf{x}} \in \mathbf{K}$. The same variant applies for 0/1 programs for which feasibility is easy to detect like e.g., for MAXCUT, k-CLUSTER or 0/1-KNAPSACK problems.

Chapter 2

Polynomial optimization

2.1 Introduction

In this thesis we consider the optimization problem:

$$\mathbf{P}: \quad f^* = \min_{\mathbf{x}} \quad \{ f(\mathbf{x}) : \mathbf{x} \in \mathbf{K} \} \tag{2.1}$$

for some given measurable function $f : \mathbb{R}^n \to \mathbb{R}$ and some Borel subset $\mathbf{K} \subset \mathbb{R}^n$. Here we insist on the fact that f^* is the *global* minimum on $\mathbf{K}$, as opposed to a *local* minimum. In full generality problem (2.1) is very difficult and there is no general purpose method, even to approximate f^*.

The primal and dual persective

If one is interested in the global minimum f^* then (2.1) has the two equivalent formulations:

$$f^* := \min_{\mu \in \mathcal{M}(\mathbf{K})_+} \{ \int_{\mathbf{K}} f d\mu \ : \ \mu(\mathbf{K}) = 1 \}, \tag{2.2}$$

and

$$f^* := \sup_{\lambda} \{ \lambda \ : \ f(\mathbf{x}) - \lambda \geq 0, \quad \forall \mathbf{x} \in \mathbf{K} \}, \tag{2.3}$$

where $\mathcal{M}(\mathbf{K})_+$ is the space of finite Borel measures on $\mathbf{K}$ (the convex positive cone of the vector space $\mathcal{M}(\mathbf{K})$ of finite signed Borel measures on $\mathbf{K}$).

That (2.3) is equivalent to (2.1) is straightforward. Indeed observe that for every

$\mathbf{x} \in \mathbf{K}$, $f(\mathbf{x}) = \int_{\mathbf{K}} f d\mu$ with $\mu \in \mathcal{M}(\mathbf{K})$ being the Dirac measure $\sigma_{\mathbf{x}}$ at $\mathbf{x}$. Therefore, the infimum in (2.2) is not larger than the infimum in (2.1). On the other hand, in (2.1) if $f^* > -\infty$ then as $f - f^* \geq 0$ on $\mathbf{K}$, integrating with respect to any probability measure $\mu \in \mathcal{M}(\mathbf{K})_+$ yields $\int_{\mathbf{K}} f d\mu \geq f^*$, which completes the proof. In addition, if $\mathbf{x}^* \in \mathbf{K}$ is a global minimizer then $\mu := \sigma_{\mathbf{x}^*}$ is also a global minimizer of (2.2).

In fact, the two formulations (2.2) and (2.3) are dual of each other in the sense of classical LP-duality if one observes that (2.2) is the infinite-dimensional LP

$$\mathbf{f}^* := \inf_{\mu \in \mathcal{M}(\mathbf{K})} \{\langle f, \mu \rangle \,:\, \langle 1, \mu \rangle = 1\,;\, \mu \geq 0\} \tag{2.4}$$

where $\langle g, \mu \rangle = \int_{\mathbf{K}} g d\mu$ for every $\mu \in \mathcal{M}(\mathbf{K})$ and every bounded measurable function g on $\mathbf{K}$ (e.g. assuming that f is bounded measurable on $\mathbf{K}$).

The optimization problems (2.3) and (2.4) are convex and even linear but also infinite-dimensional. In (2.3) one has uncountably many constraints $f(\mathbf{x}) \geq \lambda$, $\mathbf{x} \in \mathbf{K}$, whereas in (2.4) the unknown is a signed Borel measure $\mu \in \mathcal{M}(\mathbf{K})$ (and not a finite dimensional vector as in classical LP). With no other assumption on neither f nor $\mathbf{K}$, one does not known how to solve (2.3) and (2.4) because one does not have *tractable characterizations* of:

- Borel measures supported on $\mathbf{K} \subset \mathbb{R}^n$ (for solving (2.4)).
- Or functions nonnegative on $\mathbf{K}$ (for solving (2.3)).

An so in general (2.3) and (2.4) are only a *rephrasing* of (2.1)!

However, if one now considers problem (2.1) with the restrictions that:

(i) $f : \mathbb{R}^n \to \mathbb{R}$ is a polynomial (or even a semi-algebraic function), and

(ii) $\mathbf{K} \subset \mathbb{R}^n$ is a compact closed basic semi-algebraic set. (i.e. $\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geq 0, j = 1, ..., m\}$).

then one may solve (or approximate as closely as desired) the linear programs (2.3)-(2.4). This is possible due to the conjunction of two factors:

- On one side, remarkable and powerful representation theorem for polynomials positive on a basic semi-algebraic set have been produced in the nineties by real algebraic geometers, notably by Stengle [71], Schmüdgen [68], Putinar [62], and importantly, the resulting conditions can be checked by solving appropriate semidefinite programs. For mored details see e.g. [44] and [47].

- On the other side, *Semidefinite Programming* has become a central tool of convex optimization and several powerful software packages are now available to solve highly non trivial problems of relatively large size. For various classical and more recent aspects of semidefinite (and more generally conic) programming, the interested reader is referred to [26, 15].

Let us briefly explain what we mean by that. Indeed, relatively recent results from real algebraic geometry permit to characterize polynomials that are positive on $\mathbf{K}$, which is exactly what we need to solve (2.3). In addition, it turns out that those characterizations are tractable as they translate into *semidefinite* (or sometimes *linear*) conditions on the coefficients of certain polynomials that appear in some appropriate representation of the polynomial $\mathbf{x} \to f(\mathbf{x}) - \lambda$, positive on $\mathbf{K}$. Moreover, the previous representation results have a nice *dual interpret* which is concerned with sequences of reals $\mathbf{y} = (y_\alpha), \alpha \in \mathbb{N}^n$, that are the *moments* of a measure μ supported on $\mathbf{K}$, i.e.,

$$y_\alpha = \int_{\mathbf{K}} \mathbf{x}^\alpha d\mu, \quad \alpha \in \mathbb{N}^n, \tag{2.5}$$

for some Borel measure μ, which is exactly what we need to solve the linear program (2.4) when f is a polynomial. Indeed, an important observation is that in (2.1), when f is a polynomial and $f \in \mathbb{R}[\mathbf{x}]$ is written in the canonical basis $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n$, of $\mathbb{R}[\mathbf{x}]$ as

$$\mathbf{x} \to f(\mathbf{x}) = \sum_{\alpha \in \mathbb{N}^n} f_\alpha \mathbf{x}^\alpha \left(= \sum_{\alpha \in \mathbb{N}^n} f_\alpha x_1^{\alpha_1}, ..., x_n^{\alpha_n} \right),$$

for finitely many real non zero coefficients (f_α), then

$$\langle f, \mu \rangle = \int_{\mathbf{K}} f d\mu = \sum_{\alpha \in \mathbb{N}^n} f_\alpha \left(\int_{\mathbf{K}} \mathbf{x}^\alpha d\mu \right) = \sum_{\alpha \in \mathbb{N}^n} f_\alpha y_\alpha.$$

That is, the unknown measure μ in (2.2) only appears in $\int_{\mathbf{K}} f d\mu$ through its finite sequence of moments $\mathbf{y} = (y_\alpha)$ defined in (2.5).

And so in the linear program (2.4) one may replace the unknown μ with a finite sequence of real numbers $\mathbf{y} = (y_\alpha)$, along with appropriate conditions on $\mathbf{y}$ to ensure that (2.5) holds for some probability measure μ on $\mathbf{K}$. Hence, such conditions also translate into semidefinite (or sometimes linear) conditions on the $y'_\alpha s$.

Even with the above two restrictions (i)-(ii) on data expressed in terms of polynomials, problem (2.1) still encompasses a lot of important optimization problems. In particular, it includes 0/1 optimization problems, modelling $x_i \in (0,1)$ via the quadratic polynomial constraint $x_i^2 = x_i$. Therefore, one should always have in mind that one addressing a NP-hard problems in general.

In the next sections of this chapter, we present the notation used in the whole document, review some preliminary results about moments and positive polynomials and then describe the semidefinite relaxation associated with a polynomial (global) optimization problem.

2.2 Notation and definitions

Let $\mathbb{N}$ denote the set of natural numbers (including 0). The letters $x_i, y_j, x, y, z, .etc$ stand for the real valued scalars and the bold letters $\mathbf{x}, \mathbf{y}, \mathbf{z}, etc.$ stand for vectors of variables.

Denote by $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ the ring of polynomials with coefficients in $\mathbb{R}$ and with $d \in \mathbb{N}$, let $\mathbb{R}[\mathbf{x}]_d \subset \mathbb{R}[\mathbf{x}]$ be the $\mathbb{R}$-vector space of polynomials of degree at most d. For a multi-index $\alpha \in \mathbb{N}^n$, let $|\alpha| := \sum_{i=1}^n \alpha_i$ and let the notation $\mathbf{x}^\alpha$ stand for the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. The vector

$$(\mathbf{x}^\alpha)_{|\alpha| \leq d} = (1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, ..., x_1x_n, x_2^2, x_2x_3, \ldots, x_n^2, \ldots, x_1^d, \ldots, x_n^d)$$

is a (monomial) basis for vector space $\mathbb{R}[\mathbf{x}]_d$ whose dimension is $s(d) = \binom{n+d}{n}$. In this basis, a polynomial $f \in \mathbb{R}[\mathbf{x}]$ is written as

$$\mathbf{x} \mapsto f(\mathbf{x}) := \sum_{\alpha \in \mathbb{N}^n} f_\alpha \, \mathbf{x}^\alpha, \qquad f \in \mathbb{R}[\mathbf{x}],$$

for some finite vector of coefficients $\mathbf{f} = (f_\alpha) \in \mathbb{R}^{s(d)}$.

For any two real symmetric matrices $\mathbf{A}, \mathbf{B}$ the notation $\langle \mathbf{A}, \mathbf{B} \rangle$ stands for trace $(\mathbf{AB})$ and $\mathbf{A} \succeq 0$ (resp. $\mathbf{A} \succ 0$) stands for $\mathbf{A}$ is positive semidefinite (resp. positive definite).

Sums of squares

A polynomial $f \in \mathbb{R}[\mathbf{x}]$ is a sum of squares (in short SOS) if f can be written as

$$\mathbf{x} \mapsto f(\mathbf{x}) = \sum_{j \in J} f_j(\mathbf{x})^2, \qquad \mathbf{x} \in \mathbb{R}^n, \tag{2.6}$$

for some finite family of polynomials $(f_j : j \in J) \subset \mathbb{R}[\mathbf{x}]$. Denote by $\Sigma[\mathbf{x}] \subset \mathbb{R}[\mathbf{x}]$ the space of SOS polynomials.

Moment sequence

A sequence $\mathbf{y} = (y_\alpha) \subset \mathbb{R}^n$ has a representing finite Borel measure supported on set $\mathbf{K}$ if there exists a finite Borel measure μ such that

$$y_\alpha = \int_{\mathbf{K}} \mathbf{x}^\alpha d\mu \qquad \forall \alpha \in \mathbb{N}^n.$$

The *support* $\operatorname{supp} \mu$ of a finite Borel measure μ on $\mathbb{R}^n$ is the smallest (with respect to inclusion) closed set $\mathbf{K} \subseteq \mathbb{R}^n$ such that $\mu(\mathbb{R}^n \setminus \mathbf{K}) = 0$.

An old mathematical problem with many important and crucial applications is the so-called $\mathbf{K}$-moment problem, concerned with providing conditions on a real sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n$, to have a representing measure μ supported on $\mathbf{K}$. It dates back to the end of the nineteen century and has attracted attention of famous mathematicians. Among them are Stieltjies, Hausdorff, Markov, to cite a few. Crucial in the $\mathbf{K}$-moment problem is the Riesz functional associated with a sequence $\mathbf{y}$.

The Riesz functional

Given a real sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n$, the Riesz linear functional $L_{\mathbf{y}} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ associated with $\mathbf{y}$ is defined by:

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

If the sequence $\mathbf{y}$ has a representing finite Borel measure μ on $\mathbb{R}^n$, the Riesz functional $L_{\mathbf{y}}$ can be rewritten:

$$L_{\mathbf{y}}(f) \, := \sum_{\alpha \in \mathbb{N}^n} f_\alpha \, y_\alpha = \int_{\mathbb{R}^n} \left(\sum_{\alpha \in \mathbb{N}^n} f_\alpha \mathbf{x}^\alpha \right) d\mu = \int_{\mathbb{R}^n} f(\mathbf{x}) d\mu.$$

The important Riesz-Haviland theorem states necessary and sufficient conditions for a sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in N^n$, to have a representing finite Borel measure μ supported on a closed set $\mathbf{K}$.

Theorem 1. (Riesz - Haviland) *Let* $\mathbf{y} = (y_\alpha)$*,* $\alpha \in \mathbb{N}^n$*, be a real sequence and let* $\mathbf{K} \subset \mathbb{R}^n$ *be closed. Then* $\mathbf{y}$ *has a representing measure with support contained in* $\mathbf{K}$ *if and only if* $L_\mathbf{y}(f) \geq 0$ *for all polynomials* $f \in \mathbb{R}[\mathbf{x}]$ *that are nonnegative on* $\mathbf{K}$*.*

Even though it solves the $\mathbf{K}$-moment problem, Theorem 1 is still mainly theoretical in general because there is no simple characterization of polynomials nonnegative on an arbitrary closed subset $\mathbf{K}$ of $\mathbb{R}^n$. However as we will see, a notable exception is the case where the set $\mathbf{K}$ is a compact basic semi algebraic set.

Moment matrix

Given a real sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n$, the *moment* matrix $\mathbf{M}_d(\mathbf{y})$ associated with $\mathbf{y}$ is the real symmetric matrix with rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n_d$, and with entries

$$\mathbf{M}_d(\mathbf{y})(\alpha,\beta) \;:=\; L_\mathbf{y}(\mathbf{x}^{\alpha+\beta}) \;=\; y_{\alpha+\beta}, \qquad \alpha,\beta \in \mathbb{N}^n_d.$$

Equivalently, if $\mathbf{v}_d(\mathbf{x})$ is the vector of monomials $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n_d$,

$$\mathbf{M}_d(\mathbf{y}) \;=\; L_\mathbf{y}\left(\mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})^T\right),$$

where the above abuse of notation $L_\mathbf{y}\left(\mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})^T\right)$ means that the linear functional $L_\mathbf{y}$ is applied entrywise to the matrix $\mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})^T$.

If $\mathbf{y}$ has a finite representing measure μ on $\mathbb{R}^n$, and $f \in \mathbb{R}[\mathbf{x}]_d$ has coefficient vector $\mathbf{f} \in \mathbb{R}^{s(d)}$,

$$\langle \mathbf{f}, \mathbf{M}_d(\mathbf{y})\mathbf{f}\rangle \;=\; \int_{\mathbb{R}^n} f(\mathbf{x})^2\, d\mu(\mathbf{x}) \;\geq\; 0,$$

and so as $f \in \mathbb{R}[\mathbf{x}]_d$ was arbitrary, $\mathbf{M}_d(\mathbf{y}) \succeq 0$. On the other hand not every sequence $\mathbf{y}$ with $\mathbf{M}_d(\mathbf{y}) \succeq 0$ for all d, has a representing measure!

Localizing matrix

Similarly, given a real sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n$, and a polynomial $g \in \mathbb{R}[\mathbf{x}]$ with coefficient vector $\mathbf{g} = (g_\alpha)$, the *localizing* matrix $\mathbf{M}_d(g\,\mathbf{y})$ associated with $\mathbf{y}$ and g, is the real symmetric matrix with rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n_d$, and with entries

$$\mathbf{M}_d(g\,\mathbf{y})(\alpha,\beta) \;:=\; L_{\mathbf{y}}(g(\mathbf{x})\,\mathbf{x}^{\alpha+\beta}) \;=\; \sum_\gamma g_\gamma\, y_{\alpha+\beta+\gamma}, \qquad \alpha,\beta \in \mathbb{N}^n_d.$$

Equivalently, $\mathbf{M}_d(g\,\mathbf{y}) = L_{\mathbf{y}}\left(\mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})^T g(\mathbf{x})\right)$, where again the linear functional $L_{\mathbf{y}}$ is applied entry wise to the matrix $g(\mathbf{x})\mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})^T$.

If $\mathbf{y}$ has a finite representing measure μ whose support is contained in the set $\{\mathbf{x} : g(\mathbf{x}) \geq 0\}$, and if $f \in \mathbb{R}[\mathbf{x}]_d$ has coefficient vector $\mathbf{f} \in \mathbb{R}^{s(d)}$,

$$\langle \mathbf{f}, \mathbf{M}_d(\mathbf{y})\mathbf{f}\rangle \;=\; \int_{\mathbb{R}^n} f(\mathbf{x})^2\, g(\mathbf{x}) d\mu(\mathbf{x}) \;\geq\; 0,$$

and so as $f \in \mathbb{R}[\mathbf{x}]_d$ was arbitrary, $\mathbf{M}_d(g\,\mathbf{y}) \succeq 0$. On the other hand not every sequence $\mathbf{y}$ with $\mathbf{M}_d(g\,\mathbf{y}) \succeq 0$ for all d, has a representing measure supported on $\{\mathbf{x} : g(\mathbf{x}) \geq 0\}$!

2.3 Sums of squares and semidefinite programs

Checking whether a given polynomial f is nonnegative on $\mathbb{R}^n$ is very hard. On the other hand, if $f \in \mathbb{R}[\mathbf{x}]$ is SOS, i.e., $f \in \Sigma[\mathbf{x}]$, its decomposition (2.6) provides an obvious *certificate* that f is nonnegative on $\mathbb{R}^n$. And moreover, as we next see, checking whether f is in $\Sigma[\mathbf{x}]$ is easy and amounts to solving a semidefinite program. This is what makes the convex cone $\Sigma[\mathbf{x}]$ an important mathematical object that we use extensively in the sequel.

2.3.1 Semidefinite programs

Positive semidefinite matrix

Recall that for an $n \times n$ real symmetric matrix $\mathbf{M}$, the notation $\mathbf{M} \succeq 0$ means that $\mathbf{M}$ is positive semidefinite, i.e. $\mathbf{x}^T\mathbf{M}\mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. Several equivalent characterizations for $\mathbf{M} \succeq 0$ are available. For instance $\mathbf{M} \succeq 0$ if and only if:

1. $\mathbf{M} = \mathbf{V}\mathbf{V}^T$ for some $\mathbf{V} \in \mathbb{R}^n \times n$. Here matrix $\mathbf{V}$ can be chosen in $\mathbb{R}^{n\times r}$ where $r = \operatorname{rank} \mathbf{M}$.

2. $\mathbf{M} = (\mathbf{v}_j^T\mathbf{v}_j)^n_{i,j=1}$ for some vectors $(\mathbf{v}_j)^n_{j=1} \subset \mathbb{R}^n$. Here the $\mathbf{v}_j$'s can be chosen in $\mathbb{R}^r$ where $r = \operatorname{rank} \mathbf{M}$.

3. All eigenvalues of $\mathbf{M}$ are nonnegative.

Semidefinite programs

Let $\mathcal{S}_p \subset \mathbb{R}^{p\times p}$ be the space of real $p \times p$ symmetric matrices. Whenever $\mathbf{A}, \mathbf{B} \in \mathcal{S}_p$, the notation $\mathbf{A} \succeq \mathbf{B}$ (resp. $\mathbf{A} \succ \mathbf{B}$) stands for $\mathbf{A} - \mathbf{B} \succeq 0$ (resp. $\mathbf{A} - \mathbf{B} \succ 0$). Also, the notation $\langle \mathbf{A}, \mathbf{B} \rangle$ stands for trace $(\mathbf{AB})$ (= trace $(\mathbf{BA})$). In canonical form, a semidefinite program reads:

$$\mathbf{P}: \qquad \inf_{\mathbf{x}} \quad \{\mathbf{c}'\mathbf{x} : \mathbf{F}_0 + \sum_{i=1}^{n} \mathbf{F}_i x_i \succeq 0\} \tag{2.7}$$

where $\mathbf{c} \in \mathbb{R}^n$, and $\{\mathbf{F}_i\}_{i=0}^n \subset \mathcal{S}_p$ for some $p \in \mathbb{N}$. Denote by $\inf \mathbf{P}$ its optimal value.

Associated with $\mathbf{P}$ is a *dual* problem $\mathbf{D}$ which is also a semidefinite program and which reads:

$$\mathbf{D}: \qquad \sup_{\mathbf{Z}} \quad \{-\langle \mathbf{F}_0, \mathbf{Z} \rangle \quad : \quad \langle \mathbf{F}_i, \mathbf{Z} \rangle = c_i, \qquad i = 1, 2, ..., n; \quad \mathbf{Z} \succeq 0\} \tag{2.8}$$

with optimal value denoted $\sup \mathbf{D}$.

Weak duality states that $\inf \mathbf{P} \geq \sup \mathbf{D}$ and holds without any assumption. Indeed, if $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{Z} \succeq 0$ are feasible solutions of (2.7) and (2.8) respectively, then

$$\mathbf{c}'\mathbf{x} = \sum_{i=1}^{n} \langle \mathbf{F}_i, \mathbf{Z} \rangle x_i = \left\langle \sum_{i=1}^{n} \mathbf{F}_i x_i, \mathbf{Z} \right\rangle \geq -\langle \mathbf{F}_0, \mathbf{Z} \rangle.$$

The absence of a duality gap (i.e. $\inf \mathbf{P} = \sup \mathbf{D}$ is not always valid but holds under some (sufficient) strict feasibility conditions.

Theorem 2.3.1. (Strong duality) *Let* $\mathbf{P}$ *and* $\mathbf{D}$ *be as in* (2.7) *and* (2.8) *respectively.*

- *If there exists* $\mathbf{x} \in \mathbb{R}^n$ *such that* $\mathbf{F}(\mathbf{x}) \succ 0$ *then* $\inf \mathbf{P} = \sup \mathbf{D}$ *and* $\inf \mathbf{P} = \max \mathbf{D}$ *if the optimal value is finite.*

- *If there exists* $\mathbf{Z} \succ 0$ *feasible for* $\mathbf{D}$ *then* $\inf \mathbf{P} = \sup \mathbf{D}$ *and* $\min \mathbf{P} = \sup \mathbf{D}$ *if the optimal value is finite.*

- *If there exists* $\mathbf{Z} \succ 0$ *feasible for* $\mathbf{D}$ *and* $\mathbf{x} \in \mathbb{R}^n$ *such that* $\mathbf{F}(\mathbf{x}) \succ 0$ *then* $\min \mathbf{P} = \max \mathbf{D}$.

The strict feasibility condition in Theorem (2.3.1) is a specialization of Slater's condition in convex programming when applied to the conic problem $\mathbf{P}$.

An important property of semidefinite programs is their tractability. Indeed, up to arbitrary precision $\epsilon > 0$ fixed, solving a semidefinite program can be done in time polynomial in the input size of the problem.

2.3.2 Sums of squares

Recall that $\mathbf{v}_d(\mathbf{x}) = (\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n_d$, is a (monomial) basis of $\mathbb{R}[\mathbf{x}]_d$.

Theorem 2.3.2. *A polynomial* $f \in \mathbb{R}[\mathbf{x}]_{2d}$ *is a sum of squares (SOS) if and only if there exists a real symmetric and positive semidefinite matrix* $\mathbf{Q} \in \mathbb{R}^{s(d) \times s(d)}$ *such that* $f(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})'\mathbf{Q}\mathbf{v}_d(\mathbf{x})$, *for all* $\mathbf{x} \in \mathbb{R}^n$.

Proof. Suppose there exists a real symmetric $s(d) \times s(d)$ matrix $\mathbf{Q} \succeq 0$ for which $f(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})'\mathbf{Q}\mathbf{v}_d(\mathbf{x})$, for all $\mathbf{x} \in \mathbb{R}^n$. From its spectral decomposition $\mathbf{Q}$ can be writtent as $\sum_{i=1}^s \mathbf{h}_i\mathbf{h}_i'$ for some $s \in \mathbb{N}$ and some vectors $h_i \in \mathbb{R}^{s(d)}$, $i = 1, \ldots, s$. And so,

$$f(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})' \left(\sum_{i=1}^s \mathbf{h}_i\mathbf{h}_i' \right) \mathbf{v}_d(\mathbf{x}) \; = \sum_{i=1}^s \underbrace{\mathbf{v}_d(\mathbf{x})'\mathbf{h}_i}_{h_i(\mathbf{x})} \underbrace{\mathbf{h}_i'\mathbf{v}_d(\mathbf{x})}_{h_i(\mathbf{x})} = \sum_{i=1}^s h_i(\mathbf{x}))^2, \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

Conversely, suppose that $f \in \mathbb{R}[\mathbf{x}]_{2d}$ is SOS, i.e., $f(\mathbf{x}) = \sum_{i=1}^s h_i^2(\mathbf{x})$ for some finite family $(h_i) \subset \mathbb{R}[\mathbf{x}]$. Then necessarily, the degree of each h_i is bounded by d. Let $\mathbf{h}_i \in \mathbb{R}^{s(d)}$ be the vector of coefficients of the polynomial h_i, i.e., $h_i(\mathbf{x}) = \mathbf{h}_i'\mathbf{v}_d(\mathbf{x})$, $i = 1, \ldots, s$. Thus,

$$f(\mathbf{x}) = \sum_{i=1}^s \mathbf{v}_d(\mathbf{x})' \underbrace{\mathbf{h}_i\,\mathbf{h}_i'}_{\mathbf{Q}} \mathbf{v}_d(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})'\mathbf{Q}\mathbf{v}_d(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^n,$$

with $\mathbf{Q} \in \mathbb{R}^{s(d)} \times \mathbb{R}^{s(d)}$, $\mathbf{Q} := \sum_{i=1}^k \mathbf{h}_i\mathbf{h}_i \succeq 0$. □

Given a SOS polynomial $f \in \mathbb{R}[\mathbf{x}]_{2d}$, the identity $f(\mathbf{x}) = \mathbf{v}_d(\mathbf{x})'\mathbf{Q}\mathbf{v}_d(\mathbf{x}) = \langle \mathbf{Q}, \mathbf{v}_d(\mathbf{x})\mathbf{v}_d(\mathbf{x})' \rangle$ for all $\mathbf{x}$, provides linear equations that the coefficients of the matrix $\mathbf{Q}$ must satisfy.

Hence, writing

$$\mathbf{v}_d(\mathbf{x})'\mathbf{v}_d(\mathbf{x}) = \sum_{\alpha\in\mathbb{N}^n_{2d}} \mathbf{B}_\alpha\, \mathbf{x}^\alpha,$$

for appropriate real symmetric matrices $\mathbf{B}_\alpha$, checking whether the polynomial $f = \sum_\alpha f_\alpha \mathbf{x}^\alpha$ is SOS reduces to solving the semidefinite program:

$$\begin{aligned}&\text{Find } \mathbf{Q}\in\mathbb{R}^{s(d)\times s(d)} \text{ such that:}\\ &\mathbf{Q}=\mathbf{Q}',\quad \mathbf{Q}\succeq 0,\quad \langle \mathbf{Q},\mathbf{B}_\alpha\rangle = f_\alpha,\quad \forall\alpha\in\mathbb{N}^n_{2d}.\end{aligned} \tag{2.9}$$

which, as already mentioned, is a tractable convex optimization problem for which efficient software packages are available. Observe that the size $s(d) = \binom{n+d}{d}$ of the semidefinite program (2.9) is bounded by n^d.

Example. Consider the polynomial in $\mathbb{R}[x_1, x_2]$:

$$f(\mathbf{x}) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4.$$

Suppose we want to check whether f is a sum of squares. As f is homogeneous, we may attempt to find $\mathbf{Q}$ with rows and columns indexed only by the quadratic monomials x_1^2, x_1x_2, x_2^2.

$$f(\mathbf{x}) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4 \tag{2.10}$$

$$= (x_1^2 \quad x_1x_2 \quad x_2^2)\begin{pmatrix} q_{11} & q_{12} & q_{13}\\ q_{21} & q_{22} & q_{23}\\ q_{31} & q_{32} & q_{33}\end{pmatrix}\begin{pmatrix} x_1^2\\ x_1x_2\\ x_2^2\end{pmatrix} \tag{2.11}$$

$$= q_{11}x_1^4 + q_{22}x_2^4 + (q_{33}+2q_{21})x_1^2x_2^2 + 2q_{31}x_1^3x_2 + 2q_{23}x_1x_3^2, \tag{2.12}$$

for some $\mathbf{Q}\succeq 0$. Equating coefficients, we have:

$$q_{11} = 2, q_{22} = 5, q_{33} + 2q_{12} = -1, 2q_{12} = 2, q_{23} = 0.$$

One particular solution: $q_{33} = 5, q_{12} = -3$ and we have:

$$0 \preceq \mathbf{Q} = \begin{pmatrix} 2 & -3 & 1\\ -3 & 5 & 0\\ 1 & 0 & 5\end{pmatrix} = \mathcal{H}\mathcal{H}',\ \text{with } \mathcal{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 2 & 0\\ -3 & 1\\ 1 & 3\end{pmatrix}.$$

and so

$$f(x_1,x_2) = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_1x_2)^2 + \frac{1}{2}(x_2^2 + 3x_1x_2)^2,$$

which is indeed a sum of squares.

2.4 Moments and positive polynomials

Checking whether f is nonnegative on $\mathbb{R}^n$ is very hard but we have just seen that checking whether a given polynomial $f \in \mathbb{R}[\mathbf{x}]_{2d}$ is SOS reduces to solving a semidefinite program whose unknown matrix $\mathbf{Q} \succeq 0$ has size $O(n^d)$. However not all nonnegative polynomials are SOS and in fact, for fixed degree $2d$, the cone of nonnegative polynomials is much larger than that of SOS; see e.g. Blekherman [8].

However, if we are now interested in polynomials that are nonnegative on a compact set $\mathbf{K} \subset \mathbb{R}^n$, the situation is much better if $\mathbf{K}$ is a basic semi-algebraic set. Indeed from relatively recent results from real algebraic geometry, and under some conditions on the polynomials that define the compact basic semi-algebraic set $\mathbf{K}$, powerful representation for positive polynomials on $\mathbf{K}$ are available and are particularly useful for solving problem (2.3).

Moreover these representation results also have a dual facet concerned with the so-called $\mathbf{K}$-moment problem which is concerned with conditions on a given sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n$, to have a representing finite Borel measure μ supported on $\mathbf{K}$. And those results are also particular useful for solving problem (2.2). These two real algebraic and functional analysis facets nicely express the well-known duality between positive polynomials and moments, which is why if the former are useful for solving (2.3) then naturally and not surprisingly, the latter are useful for solving the dual problem (2.2).

Given polynomials $g_1, \ldots, g_m \in \mathbb{R}[\mathbf{x}]$, the set

$$\mathbf{Q}(g) = \mathbf{Q}(g_1, \ldots, g_m) := \left\{ u_0 + \sum_{j=1}^m u_j\, g_j \;:\; u_0, u_j \in \Sigma[\mathbf{x}] \right\}, \tag{2.13}$$

is a convex cone called the *quadratic module* generated by the family $g_1, g_2, \ldots, g_m$.

Assumption 1. *With $(g_j)_{j=1}^m \subset \mathbb{R}[\mathbf{x}]$, there exists $N \in \mathbb{N}$ such that the quadratic polynomial $\mathbf{x} \to N - ||\mathbf{x}||^2$ belongs to $\mathbf{Q}(g)$.*

When Assumption 1 holds the quadratic module $\mathbf{Q}(g)$ is said to be *Archimedean.*

Observe that Assumption 1 is not on $\mathbf{K}$ itself but on its representation via the g_j's. It is not very restrictive. For instance, it is satisfied in the following cases:

. $\mathbf{K}$ is compact and all the $g_j's$ are affine (and so $\mathbf{K}$ is a convex polytope). Or

. The set $\{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geq 0\}$ is compact for some $j \in \{1, \ldots, m\}$.

Moreover, if $\mathbf{K}$ is compact, it is contained in a ball $\{\mathbf{x} : M - \|\mathbf{x}\|^2 \geq 0\}$ for some M. If M is known then it suffices to add the redundant constraint $g_{m+1}(\mathbf{x}) \geq 0$ (with $\mathbf{x} \mapsto g_{m+1}(\mathbf{x}) = M - \|\mathbf{x}\|^2$) in the definition of $\mathbf{K}$ and Assumption 1 holds.

Theorem 2.4.1 (Putinar's Positivstellensatz [62]). *Let $\mathbf{K} \subset \mathbb{R}^n$ be the basic semi-algebraic set:*

$$\mathbf{K} := \{\, \mathbf{x} \in \mathbb{R}^n \;:\; g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, m \,\} \tag{2.14}$$

for some $(g_j) \subset \mathbb{R}[\mathbf{x}]$, and let Assumption 1 hold. If $f \in \mathbb{R}[\mathbf{x}]$ is strictly positive on $\mathbf{K}$ then $f \in \mathbf{Q}(g)$, that is,

$$f \;=\; \sigma_0 + \sum_{j=1}^{m} \sigma_j \, g_j, \tag{2.15}$$

for some SOS polynomials $\sigma_j \in \Sigma[\mathbf{x}]$, $j = 0, 1, \ldots, m$.

In Theorem 2.4.1 nothing is said on the degree of the SOS polynomials σ_j involved in the representation (2.15) of f. This is the reason why Theorem 2.4.1 does not lead to a polynomial time algorithm to check whether a polynomial f is positive on $\mathbf{K}$.

Observe that the SOS polynomials σ_j in (2.15) provide an obvious *certificate* of non negativity on $\mathbf{K}$ for f. The strength of Putinar's Positivstellensatz is to state the (almost) converse result, i.e., that every polynomial f strictly positive on $\mathbf{K}$ has such a certificate.

The next theorem is a dual facet of Theorem (2.4.1).

Theorem 2.4.2 (Dual facet of Theorem 2.4.1). *Let $\mathbf{K} \subset \mathbb{R}^n$ be the basic semi-algebraic set defined in (2.14) for some $(g_j) \subset \mathbb{R}[\mathbf{x}]$, and let Assumption 1 hold. A sequence $\mathbf{y} = (y_\alpha)$ has a representing measure μ supported on $\mathbf{K}$ if and only if, for all $d = 0, 1, \ldots$*

$$\mathbf{M}_d(\mathbf{y}) \succeq 0; \qquad \mathbf{M}_d(g_j\,\mathbf{y}) \succeq 0, \quad j = 1, \ldots, m$$

or equivalent,

$$L_{\mathbf{y}}(f) \geq 0; \qquad L_{\mathbf{y}}(g_j\, f) \geq 0, \quad \forall f \in \Sigma[\mathbf{x}].$$

Proof. For every $j \in \{1, ..., m\}$ and every $f \in \mathbb{R}[\mathbf{x}]_d$, the polynomial $f^2 g_j$ is nonnegative on $\mathbf{K}$. Therefore if $\mathbf{y}$ is the moment sequence of a measure μ supported on $\mathbf{K}$, then $\int_{\mathbf{K}} f^2 g_j d\mu \geq 0$. Equivalently, $L_{\mathbf{y}}(f^2\, g_j) \geq 0$ or in view of the definition of the localizing matrix, $\mathbf{M}_d(g_j\, \mathbf{y}) \succeq 0$ for all $d = 0, 1, \ldots$

Conversely, assume that $\mathbf{M}_d(g_j\, \mathbf{y}) \succeq 0$ for all $d = 0, 1, \ldots$ and $j = 1, \ldots, m$. Equivalently, $L_{\mathbf{y}}(f^2\, g_j) \geq 0$ for all $f \in \mathbb{R}[\mathbf{x}]$. If we prove that $L_{\mathbf{y}}(f) \geq 0$ for all polynomials nonnegative on $\mathbf{K}$ then by Theorem 1 $\mathbf{y}$ has a representing measure supported on $\mathbf{K}$. Let $f > 0$ on $\mathbf{K}$. As $\mathbf{K}$ is compact and Assumption 1 holds, by Theorem 2.4.1, $f = \sigma_0 + \sum_{j=1}^m \sigma_j g_j$, for some SOS polynomials $\sigma_j \in \Sigma[\mathbf{x}]$, $\quad j = 0, 1, \ldots, m$. But then

$$L_{\mathbf{y}}(f) \;=\; L_{\mathbf{y}}(\sigma_0) + \sum_{j=1}^m L_{\mathbf{y}}(\sigma_j\, g_j) \;\geq\; 0.$$

Next, if $f \in \mathbb{R}[\mathbf{x}]$ is only nonnegative on $\mathbf{K}$ then for arbitrary $\epsilon > 0$, $f + \epsilon > 0$ on $\mathbf{K}$, and so $L_{\mathbf{y}}(f + \epsilon) = L_{\mathbf{y}}(f) + \epsilon y_0 \geq 0$. As $\epsilon > 0$ was arbitrary, we obtain $L_{\mathbf{y}}(f) \geq 0$. □

In the next section we will see that Theorem 2.4.1 or its dual facet Theorem 2.4.2 are crucial in the proof of convergence of a numerical scheme for approximating as closely as desired from below the optimal value f^* of problem (2.1).

2.5 Semidefinite relaxations for global optimization

In this section we introduce the moment-SOS approach solving the fundamental (global) optimization problem (2.1):

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

where $f \in \mathbb{R}[\mathbf{x}]$ and $\mathbf{K} \subset \mathbb{R}^n$ is the basic semi-algebraic set defined in (2.14) for some $(g_j) \subset \mathbb{R}[\mathbf{x}]$.

We assume that Assumption (1) holds. Recall that solving problem (2.1) is equivalent to solving the infinite-dimensional linear program (LP) (2.2) or its LP dual (2.3) and by the hierarchy of semidefinite relaxations, we can solving approximately (and sometimes exactly) these problems LPs (2.2) or (2.3). The approach in this section was outline in [43, 44].

2.5.1 Semidefinite relaxations approach

We next see that the optimal value f^* of problem (2.1) can be approximated as closely as desired (and sometimes can be obtained exactly) by solving an appropriate hierarchy of *convex relaxations* of (2.1), each convex relaxation in the hierarchy is a semidefinite program whose size increases with its rank in the hierarchy.

Let $v_0 := \lceil(\deg f)/2\rceil$, $v_j := \lceil(\deg g_j)/2\rceil$, $j = 1,\ldots,m$, and for every $d \geq \max_{j=0,\ldots,m} v_j$ consider the semidefinite program:

$$\begin{aligned}
\rho_d := \quad & \inf_{\mathbf{y}} \; L_{\mathbf{y}}(f) \\
\text{s.t} \quad & \mathbf{M}_d(\mathbf{y}) \succeq 0 \\
& \mathbf{M}_{d-v_j}(g_j\mathbf{y}) \succeq 0, \quad j = 1,\ldots,m \\
& L_{\mathbf{y}}(1) = 1.
\end{aligned} \tag{2.16}$$

The dual of problem (2.16) is the semidefinite program:

$$\begin{aligned}
\rho_d^* := \quad & \sup_{\lambda,\{\sigma_j\}} \; \lambda \\
\text{s.t} \quad & f - \lambda = \sigma_0 + \sum_{j=1}^m \sigma_j \, g_j \\
& \deg \sigma_0 \leq 2d;\ \deg \sigma_j \leq 2d - 2v_j, \quad j = 1,\ldots,m.
\end{aligned} \tag{2.17}$$

Note that if in the definition of $\mathbf{K}$ there is an equality constraint $g_j(\mathbf{x}) = 0$, then one has the equality constraint $\mathbf{M}_{i-v_j}(g_j\mathbf{y}) = 0$ in (2.16) and accordingly, in (2.17) the polynomial $\sigma_j \in \mathbb{R}[\mathbf{x}]$ is not required to be a SOS.

Interpretation: Problem (2.16) is an obvious relaxation of the LP problem (2.2). To see this recall that under Assumption 1 the set $\mathbf{K}$ is compact and so if μ is a probability measure on $\mathbf{K}$ (hence a feasible solution of (2.2)) all its moments $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n$, are well-defined, and so $\mathbf{M}_d(\mathbf{y}), \mathbf{M}_d(g_j\,\mathbf{y}) \succeq 0$, $j = 1,\ldots,m$. That is, $\mathbf{y}$ is a feasible solution of (2.16). Moreover $L_{\mathbf{y}}(f) = \int_{\mathbf{K}} f d\mu \geq f^*$ and so $\rho_d \leq f^*$ for every $d \geq \max_{0,\ldots,m} v_j$. Indeed the conditions $\mathbf{M}_d(\mathbf{y}) \succeq 0$ and $\mathbf{M}_{d-v_j}(g_j\,\mathbf{y}) \succeq 0$ are only necessary conditions for $\mathbf{y}$ to have a representing measure on $\mathbf{K}$.

In the same vein, for every feasible solution $((\sigma_j), \lambda)$ of (2.17), the polynomial $\mathbf{x} \mapsto f(\mathbf{x}) - \lambda$ being in Putinar form (2.15), is obviously nonnegative on $\mathbf{K}$. Therefore $\lambda \leq f^*$. And so one retrieves that if (2.16) is a relaxation of (2.2), then by duality (2.17) is a strengthening of (2.3).

Of course the size of (2.16) increases with d in the hierarchy because as one considers more and more moments y_α, the matrix $\mathbf{M}_d(\mathbf{y})$ has $s(d)$ rows and columns. The following theorem states the convergence of the above hierarchy of relaxations (2.16).

Theorem 2.5.1 (**Convergence theorem**). *Let Assumption* (1) *hold and consider the hierarchy of semidefinite relaxations* (2.16) *with optimal value* ρ_d.

(i) *The sequence* (ρ_d), $d \in \mathbb{N}$, *is monotone nondecreasing and* $\rho_d \to f^*$ *as* $d \to \infty$.

(ii) *Assume that* (2.1) *has a unique optimal solution* $\mathbf{x}^* \in \mathbf{K}$ *and let* $\mathbf{y}^d$ *be a nearly optimal solution of* (2.16), *e.g. with value* $\mathbf{L}_{\mathbf{y}^d}(f) \leq \rho_i + 1/i$. *Then as* $d \to \infty, \mathbf{L}_{\mathbf{y}^d}(x_j) \to x_j^*$ *for every* $j = 1, \ldots, n$.

(iii) *If* $\mathbf{K}$ *is a finite set defined by polynomials equality and inequality constraints then finite convergence* $\rho_d \to f^*$ *takes place.*

A detailed proof can be found in e.g. [43, 44].

2.5.2 Extraction of global minimizers

After solving the semidefinite relaxation (2.16) for some value of $d \in \mathbb{N}$, two important issues remain to be investigated:

i. Can we decide whether $\rho_d < f^*$ or $\rho_d = f^*$?

ii. If $\rho_d = f^*$, can we *extract* at least one global minimizer $x^* \in \mathbf{K}$ from an optimal solution $\mathbf{y}$ of the semidefinite relaxation (2.16)?

An easy case is when (2.16) has an optimal solution $\mathbf{y}^*$ which satisfies rank $\mathbf{M}_d(\mathbf{y}^*) = 1$ so that neccessarily $\mathbf{M}_d(\mathbf{y}^*) = \mathbf{v}_d(\mathbf{x}^*)\mathbf{v}_d(\mathbf{x}^*)'$ for some $\mathbf{x}^* \in \mathbb{R}^n$. That is, $\mathbf{y}^*$ is the vector of moments up to order $2d$ of the Dirac measure at some point $\mathbf{x}^* \in \mathbb{R}^n$. In particular $\rho_d = L_{\mathbf{y}}(f) = f(\mathbf{x}^*)$ and the constraints $\mathbf{M}_{d-v_j}(g_j\,\mathbf{y}^*) \succeq 0$, $j = 1, \ldots, m$, imply $\mathbf{x}^* \in \mathbf{K}$. Therefore from $\rho_d = f(\mathbf{x}^*) \leq f^*$ we conclude that $\mathbf{x}^*$ is a global minimizer of (2.1).

When rank $\mathbf{M}_d(\mathbf{y}^*) \neq 1$ the following sufficient condition is very helpful to extract multiple global minimizers for problem (2.1):

Theorem 2.5.2 (**Rank-test theorem**, see [13]). *Let* $f \in \mathbb{R}[\mathbf{x}]$ *and suppose that the optimal value* ρ_d *of problem* (2.16) *is attained at some optimal solution* $\mathbf{y}^*$. *Let*

$v := \max_{j=1,\ldots,m} v_j$. *If rank* $\mathbf{M}_{d-v}(\mathbf{y}^*) = \mathbf{M}_d(\mathbf{y}^*)$ *then* $\rho_d = f^*$ *and there are at least* $s := \operatorname{rank} \mathbf{M}_d(\mathbf{y}^*)$ *global minimizers.*

A detailed proof can be found in e.g. [13] and the implementation of can be found in [43] .

When the rank condition of Theorem (2.5.2) is satisfied, rank $\mathbf{M}_d(\mathbf{y}^*)$ global minimizers are encoded in the optimal solution $\mathbf{y}^*$ of (2.16) and they can be extracted by a linear algebra procedure described in [28] and implemented in the GloptiPoly software [29].

Finally, the moment-SOS algorithm for solving (or at least approximate) the polynomial optimization problem (2.1) reads as follows:

Algorithm 2.5.3.

Input: A polynomial $f \in \mathbb{R}[\mathbf{x}]$ of degree $2v_0$ or $2v_0 - 1$; A set $\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geq 0, j = 1, \ldots, m\}$, where the polynomials $g_j \in \mathbb{R}[\mathbf{x}]$ are of degree $2v_j$ or $2v_j - 1$, $j = 1, ..., m$; A number k, the order of the highest relaxation that can be solved.

Output: The optimal value $f^* = \min_{\mathbf{x}} \{ f(\mathbf{x}) \, : \, \mathbf{x} \in \mathbf{K} \}$ and a list of global minimizers or a lower bound ρ_k on f^*.

Algorithm:

1. Solve the semidefinite program (2.16) with optimal value ρ_d and optimal solution $\mathbf{y}^*$ (if it exists).

2. If there is no optimal solution $\mathbf{y}^*$, (2.16) only provides a lower bound $\rho_d \leq f^*$. If $d < k$, then set $d := d + 1$ and go to Step 1; Otherwise stop and output ρ_k.

3. If rank $\mathbf{M}_{d-v}(\mathbf{y}^*) = \mathbf{M}_d(\mathbf{y}^*)$ (with $v := \max_j v_j$) then $\rho_d = f^*$ and there are at least rank $\mathbf{M}_d(\mathbf{y}^*)$ global minimizers. Extract the solutions[1].

4. If rank $\mathbf{M}_{d-v}(\mathbf{y}^*) \neq \operatorname{rank} \mathbf{M}_d(\mathbf{y}^*)$ and $d < k$ then set $d := d + 1$ and go to Step 1. Otherwise stop and output ρ_k which is a lower bound on f^*.

Algorithm 2.5.3 with a linear algebra procedure for extracting global minimizers (at step 3) is implemented in the GloptiPoly software described in [29].

Example 2.5.4. Consider the optimization problem

$$f^* \;=\; \min_{\mathbf{x}} \{ \, x_1^2 x_2^2 \, (x_1^2 + x_2^2 - 1) \; : \; x_1^2 + x_2^2 \, \leq \, 4 \, \}$$

[1] For instance by the linear algebra procedure described in [28] and implemented in the GloptiPoly software [29].

whose optimal value is $f^* = -1/27$ with global minimizers $\mathbf{x}^* = (\pm\sqrt{3}/3, \pm\sqrt{3}/3)$. Running the GloptiPoly software, the optimal value f^* is obtained at $d = 4$ (up to some numerical imprecision) and the rank test is passed. The four (approximately) optimal solutions $(\pm 0.5774, \pm 0.5774) \approx \mathbf{x}^*$ are extracted.
If one adds the additional nonconvex constraint $x_1 x_2 \geq 1$, the optimal value is obtained with $\rho_3 \approx 1$ and two optimal solutions $(x_1^*, x_2^*) = (-1, -1)$ and $(x_1^*, x_2^*) = (1, 1)$ are extracted.

2.6 Discussion

2.6.1 Alternative LP-relaxations

The moment-SOS approach implemented in Algorithm 2.5.3 is based on Putinar's Positivstellenstaz (Theorem 2.4.1), an extremely powerful representation result. However other representation results of real algebraic geometry are available. For instance, Stengle's Positivstellensatz [71] is even more powerful as it applies to sets $\mathbf{K}$ that are much more general than compact basic semi-algebraic sets as $\mathbf{K}$ described in (2.14), and to polynomials that are only nonnegative on $\mathbf{K}$ (and not strictly positive on $\mathbf{K}$ as in Theorem 2.4.1); however it is not easily amenable to practical implementation.

On the other hand, another representation result due to Krivine [39] (and also Handelman [25] for representation on polytopes) is also amenable to practical computation. Let $\mathbf{K} \subset \mathbb{R}^n$ be the compact basic semi-algebraic set defined in $\mathbf{K}$, and assume that the polynomials g_j, $j = 0, \ldots, m$ (with $g_0 = 1$), generate the algebra $\mathbb{R}[\mathbf{x}]$. For sake of simplicity assume that (possibly after scaling) $0 \leq g_j \leq 1$ on $\mathbf{K}$ for all $j = 1, \ldots, m$. If $f \in \mathbb{R}[\mathbf{x}]$ is strictly positive on $\mathbf{K}$ then

$$f(\mathbf{x}) = \sum_{\alpha,\beta \in \mathbb{N}^m} c_{\alpha\beta} \left(\prod_{j=1}^{m} g_j(\mathbf{x})^{\alpha_j} \right) \left(\prod_{j=1}^{m} (1 - g_j(\mathbf{x}))^{\beta_j} \right) \qquad \forall \mathbf{x} \in \mathbb{R}^n, \tag{2.18}$$

for finitely many real nonnegative coefficients $(c_{\alpha\beta})$. See e.g. [44].

Notice that detecting whether f can be written as in (2.18) with say $|\alpha| + |\beta| \leq d$, reduces to solving a linear system of the form $\mathbf{A}\mathbf{c} = \mathbf{b}$, $\mathbf{c} \geq 0$, which can be done by solving a linear program. Indeed, as (2.18) has to hold for every $\mathbf{x} \in \mathbb{R}^n$, expanding (in the monomial basis $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n$) polynomials in each side of "=" in (2.18) and equating coefficients, yields finitely many *linear* constraints between the nonnegative coefficients $c_{\alpha\beta}$. Hence one may replace the semidefinite relaxations (2.16) with the

LP-relaxations

$$\ell_d = \max_{\lambda,\mathbf{c}} \{\lambda \,:\, f-\lambda = \sum_{\alpha,\beta\in\mathbb{N}^m} c_{\alpha\beta}\left(\prod_{j=1}^m g_j^{\alpha_j}\right)\left(\prod_{j=1}^m (1-g_j)^{\beta_j}\right);\quad \mathbf{c}\geq 0\} \tag{2.19}$$

and by Krivine' result [39], $\ell_d \to f^*$ as $d\to\infty$.

One might indeed prefer solving the LP-relaxations (2.19) rather than the semidefinite relaxations (2.16) (or their dual (2.17)) because LP-solvers can handle linear programs of potentially huge size whereas this is not the case for semidefinite solvers. In particular, in view of the present status of semidefinite solvers available, only problems (2.1) of small to moderate size can be solved (or approximated) by the moment-SOS approach. Unfortunately, and as explained in [44], the LP-relaxations (2.19) can never be exact:

- as soon as there is a global minimizer $\mathbf{x}^*\in\mathbf{K}$ in the interior of $\mathbf{K}$,

- or when $\mathbf{x}^*\in\partial\mathbf{K}$, if there is a non-optimal feasible solution $\mathbf{x}\in\mathbf{K}$ with $J(\mathbf{x}) = J(\mathbf{x}^*)$ (where $J(\mathbf{x}) := \{j\in\{1,\ldots,m\} \,:\, g_j(\mathbf{x})=0\}$ is the set of constraints $g_j(\mathbf{x})\geq 0$ that are active at $\mathbf{x}$).

This is very annoying since the above situation is generic for easy convex problems! Moreover the LP-relaxations (2.19) are numerically ill-conditioned as soon as d gets relatively large (especially in view of large binomial coefficients that appear in the expansion of $(1-g_j(\mathbf{x}))_j^\beta$ when β_i is relatively large).

2.6.2 Putinar versus Karush-Kuhn-Tucker

Let Assumption 1 hold for the basic semi-algebraic set $\mathbf{K}$ in (2.14) and assume that the polynomial $f-f^*$ can be written in Putinar's form

$$f-f^* \;=\; \sigma_0^* + \sum_{j=1}^m \sigma_j^*\, g_j, \tag{2.20}$$

for some SOS polynomials $(\sigma_j^*)\subset\Sigma[\mathbf{x}]$ (which is not always guaranteed as $f-f^*$ is only nonnegative on $\mathbf{K}$ and not strictly positive as required in Theorem 2.4.1). Then if $\mathbf{x}^*\in\mathbf{K}$ is a global minimizer, using that all σ_j^* are SOS,

- evaluating both sides of (2.20) at $\mathbf{x}=\mathbf{x}^*$ yields $\underbrace{\sigma_j^*(\mathbf{x}^*)}_{\lambda_j^*\geq 0}\, g_j(\mathbf{x}^*) \;=\; 0$ for every

$j = 1, \ldots, m$, and differentiating both sides of (2.20) at $\mathbf{x} = \mathbf{x}^*$ yields

$$\nabla f(\mathbf{x}^*) \;=\; \sum_{j=1}^{m} \underbrace{\sigma_j^*(\mathbf{x}^*)}_{\lambda_j^* \geq 0} \, \nabla g_j(\mathbf{x}^*),$$

- $H_f(\mathbf{x}) := f(\mathbf{x}) - f^* - \sum_{j=1}^{m} \sigma_j^*(\mathbf{x})\, g_j(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$, and $H_f(\mathbf{x}^*) = 0$.

In other words, letting $\lambda_j^* := \sigma_j^*(\mathbf{x}^*)$, $j = 1, \ldots, m$, the pair $(\mathbf{x}^*, \lambda^*) \in \mathbf{K} \times \mathbb{R}_+^m$ satisfies the Karush-Kuhn-Tucker (KKT) optimality conditions and in addition $\mathbf{x}^*$ is a global minimizer of the *extended* Lagrangian H_f. Notice that the extended Lagrangian H_f has SOS multipliers $\sigma_j^* \in \Sigma[\mathbf{x}]$ instead of scalars multipliers $\lambda_j^* \in \mathbb{R}_+$ in the usual Lagrangian $L_f := f - f^* - \sum_j \lambda_j^* \, g_j$.

By Putinar's representation (2.20) (when it holds), every global minimizer $\mathbf{x}^* \in \mathbf{K}$ of f on $\mathbf{K}$ is also a global minimizer of the extended Lagrangian H_f on the whole space $\mathbb{R}^n$. This is to contrast with the KKT conditions where $\mathbf{x}^*$ is a global minimizer of L_f on $\mathbb{R}^n$ only in the convex case, i.e., when f and $-g_j$ are convex, $j = 1, \ldots, m$.

Moreover consider a constraint $g_j(\mathbf{x}) \geq 0$ which is not active at a global minimizer $\mathbf{x}^* \in \mathbf{K}$ but which is important for f^* meaning that if it is removed from the definition (2.14) of $\mathbf{K}$, the optimal value of the resulting modified optimization problem decreases strictly. In the KKT optimality conditions the corresponding Lagrange multiplier λ_j^* vanishes whereas in Putinar's representation (2.20) $\sigma_j^*(\mathbf{x}^*) = 0\,(= \lambda_j^*)$ but σ_j^* is *not* trivial (otherwise f^* would still be a global minimum when the constraint $g_j \geq 0$ is removed). Hence $\sigma_j^* \neq 0$ in (2.20) (even if $\sigma_j(\mathbf{x}^*) = 0$) reflects the importance of the constraint $g_j(\mathbf{x}) \geq 0$, which does not happen in the KKT optimality conditions. This is due to the fact that Putinar's representation (2.20) is global.

2.6.3 The no free lunch rule

As we have seen, the moment-SOS approach is very powerful as one may approximate as closely as desired the optimal value f^* for the class of polynomial optimization problems whose feasible set $\mathbf{K}$ is a compact basic semi-algebraic set as defined in (2.14) (and for which Assumption (1) holds). As such problems encompass very hard optimization problems (including 0/1 programs) the no-free lunch rule should apply.

And indeed the reader will have noticed that in Theorem 2.5.1 nothing is said on the rate of convergence of $\rho_d \to f^*$. In fact, in Theorem 2.4.1 there exist degree bounds on the SOS polynomials σ_j in terms of the coefficients of the g_j's, the norm of f and the minimum of f on $\mathbf{K}$ (see [59]). But as expected, these bounds are not useful to analyze the computational complexity of Algorithm 2.5.3.

However, from many numerical experiments on non trivial examples (including problems with non convex or non connected, or discrete feasible sets $\mathbf{K}$), it seems that in general convergence if fast and very often even finite. This is fortunate because when d is fixed, the semidefinite relaxation (2.16) has $O(n^{2d})$ variables y_α and the moment matrix $\mathbf{M}_d(\mathbf{y})$ has size $s(d) \times s(d)$ (recall that $s(d) = O(n^d)$). So with fixed d, and even if solving the semidefinite relaxation (2.16) can be done in time polynomial in the input size of (2.16) (up to arbitrary fixed precision), the size grows rapidly with n and except for small to moderate size problems (2.1), solving (2.16) it is out of reach for all present semidefinite solvers available.

On the other hand, large scale optimization problems usually exhibit some special structure. For instance some structured sparsity and/or symmetries are often present in the data that describe the problem. In this case one may take advantage of such specific properties right from the begining and define appropriate semidefinite relaxations in the same vein as those described in (2.16) but of much smaller size. For more details and results in this direction the interested reader is refereed to e.g. [35, 37, 76, 77] for semidefinite relaxations taking advantage of symmetries and [35, 42] for sparse analogues of (2.16) in case of structured sparsity. For the latter relaxations the authors of [77] could solve problems (2.1) with up to 1000 variables!

2.7 Conclusion

From what precedes one may see that the semidefinite relaxations (2.16) are very powerful to approximate the optimal value f^* and sometimes to obtain f^* exactly as well as and some global minimizers $\mathbf{x}^* \in \mathbf{K}$. However, and unless some structured sparsity and/or symmetries can be taken into account, so far the moment-SOS approach is limited to small to medium size problems only.

In many cases of interest one may implement the first or second relaxation of the hierarchy (2.16) but not higher order relaxations because of the present limitation of semidefinite solvers. Therefore one is left with only a lower bound $\rho_d \leq f^*$ (where d is the last step in the hierarchy where the semidefinite program (2.16) can be solved). And so a natural issued one is faced with is:

Can we use an optimal solution of the semidefinite relaxation (2.16) at step d (or *a slight modification of (2.16)) to find an approximate solution of (2.1)?*, or

Can we use the moment-SOS methodology and its associated hierarchy of semidefinite relaxations (2.16) to help solve potentially large scale optimization problems (2.1)?

The purpose of this thesis is an attempt to provide answers to the above two questions.

Chapter 3

Polynomial convex underestimators

3.1 Introduction

In this chapter we consider a polynomial optimization problem $\mathbf{P}$ in the form:

$$\begin{aligned} \mathbf{P}: \quad f^* = \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t} \quad & g_j(\mathbf{x}) \geq 0, \quad j = 1, ..., m \\ & \mathbf{x} \in \mathbf{B} = [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n \\ & x_i \in \mathbb{Z}, \qquad i \in I \subseteq \{1, ..., n\}. \end{aligned} \tag{3.1}$$

where f, g_j are polynomials $\forall j = 1, ..., m$ and $\mathbf{x}^l, \mathbf{x}^u \in \mathbb{R}^n$ define the box $[\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n$. When $I \neq \emptyset$ problem $\mathbf{P}$ is called a *Mixed Integer Non Linear Program* (MINLP) because it contains both continuous and discrete variables.

For solving $\mathbf{P}$ one can use the moment-SOS approach described in Chapter 1 to compute, or at least approximate as closely as desired, the optimal value f^* and if possible to also obtain a global minimizer $\mathbf{x}^*$. However, in the hierarchy of semidefinite relaxations (2.16) associated with the moment-SOS approach, the size of semidefinite relaxations grows rapidly with the rank in the hierarchy and with the number of variables in $\mathbf{P}$. Indeed recall that typically if $\mathbf{P}$ is an optimization in $\mathbb{R}^n$ then the k-th semidefinite relaxation has $O((2k)^n)$ variables if n is fixed and $O(n^{2k})$ variables if k is fixed. Moreover it involves semidefiniteness constraints of matrices with size $O(k^n)$ if n is fixed (resp. $O(n^k)$ if k is fixed). So, especially in view of the present status of semidefinite solvers, solving such semidefinite relaxations becomes rapidly impossible unless one may exploit some symmetry and/or sparsity that appear(s) in the description of $\mathbf{P}$. Therefore if $\mathbf{P}$ has no particular structure (e.g. no symmetry or no structured sparsity that can be exploited) the moment-SOS approach is limited to problems of moderate size only. Consequently the following issue is natural and worth investigating:

"Can the moment-SOS approach be still useful (and how) to help solve potentially large scale problems?"

The purpose of this chapter is precisely to provide a positive answer to this question. Indeed we apply the moment-SOS approach to improve some deterministic method that try to solve large scale MINLP optimization problems in the format (3.1).

More precisely: For practical efficiency, a typical approach to solve or approximate a non convex MINLP problem $\mathbf{P}$ as defined in (3.1) is to first provide:

- a *convex under estimator* $\tilde{f}$ of f on $\mathbf{B}$, as tight as possible, if f is non convex, and

- a convex underestimator $-\tilde{g}_j$ of $-g_j$ on $\mathbf{B}$ if $-g_j$ is non convex, $\forall j = 1, ..., m$.

Observe that the resulting set $\{\mathbf{x} \in \mathbf{B} : \tilde{g}_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, m\}$ is convex and contains the original feasible set in (3.1). Therefore the optimization problem:

$$\begin{aligned} \mathbf{P}: \quad \tilde{f}^* = \min_{\mathbf{x}} \quad & \tilde{f}(\mathbf{x}) \\ \text{s.t} \quad & \tilde{g}_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, m \\ & \mathbf{x} \in \mathbf{B} = [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n. \end{aligned} \tag{3.2}$$

is obviously a *relaxation* of $\mathbf{P}$ and so $\tilde{f}^*$ is a lower bound on f^*. Moreover, being convex, the optimization problem (3.2) can be solved efficiently via standard algorithms of convex optimization. This technique can be used to compute lower bounds in a *Branch and Bound* (B&B) procedure where a certain box $\mathbf{B}_\alpha$ is associated with each node α of the search tree. A lower bound is computed at each node α by solving a convex problem as defined in (3.2) with $\mathbf{B} := \mathbf{B}_\alpha$.

In particular, for MINLP problems where some $0/1$ (and more generally, discrete) variables $x_j \in \mathbb{Z}$, $j \in J \subseteq \{1, \ldots, n\}$, are present, the integrality constraints are relaxed to the convex box constraint $\mathbf{x} \in \mathbf{B}$. Such lower bounding techniques are already implemented in the B&B search treee for Mixed Integer Linear Programs (MILPs) of powerful softwares such as XPRESS-MP or CPLEX, to cite a few. For

MINLPs, an example of a popular B&B method is the so-called α**BB** algorithm and its variants; see [1, 4, 6, 3, 22, 23].

Of course, as the reader may guess, crucial for the overall performance of the algorithm is the quality of the lower bounds computed at each node of the B&B search tree. And in addition, the quality of the lower bounds is strongly dependent on the quality of the convex under estimators of f and $-g_j$. Therefore developing *tight* convex underestimators for non convex polynomials on a box $\mathbf{B}$ is of crucial importance for B&B procedures, and for the α**BB** method in particular.

The context. We consider polynomials $f \in \mathbb{R}[\mathbf{x}]$ of a few variables (to fix ideas say e.g. $n \leq 5$) only. Why? Because in typical large scale MINLPs, when the function f is a non convex polynomial it is a sum $\sum_\ell f_\ell$ of polynomials (or monomials) f_ℓ with a few variables only; for instance if $f \in \mathbb{R}[\mathbf{x}]_d$ then f is a weighted sum of monomials $\mathbf{x}^\alpha$, each containing less than d variables, and in most MINLPs d is rather small. Similarly, for a non convex polynomial constraint $g_j(\mathbf{x}) \leq 0$, the polynomial g_j is also concerned with a few variables only and the coupling of all variables is done through the intersection of the levels sets $\{\mathbf{x} : g_j(\mathbf{x}) \leq 0\}$, $j = 1, \ldots, m$. And so for practical efficiency (e.g. in the α**BB** method), a convex under estimator $\tilde{f}$ of f is simply the sum $\sum_\ell \tilde{f}_\ell$ where each $\tilde{f}_\ell$ is a convex underestimator of f_ℓ which can be computed efficiently.

Contribution. In this chapter we consider the generic problem of computing *polynomial* convex underestimators for a non convex polynomial $f \in \mathbb{R}[\mathbf{x}]$ on a box $\mathbf{B} = [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n$. As already mentioned above, such a problem is encountered when one wishes to compute a convex underestimator $\tilde{f}$ of an objective function f, or when one wishes to relax a nonconvex constraint $g(\mathbf{x}) \leq 0$ to a convex constraint $\tilde{g}(\mathbf{x}) \leq 0$.

More precisely, we are looking for a convex polynomial $f_d \in \mathbb{R}[\mathbf{x}]_d$ (with degree d fixed) which approximates f from below on a given box $\mathbf{B} \subset \mathbb{R}^n$. Hence a polynomial candidate f_d must satisfy two major conditions:

- The **nonnegativity** condition "$f \geq f_d$ on $\mathbf{B}$".
- The **convexity** condition on $\mathbf{B}$, i.e., the Hessian matrix $\nabla^2 f_d$ must be positive semidefinite (i.e., $\nabla^2 f_d \succeq 0$) on $\mathbf{B}$.

But of course there are many potential polynomial candidates $f_d \in \mathbb{R}[\mathbf{x}]_d$ and therefore a meaningful criterion to select the "best" among them is essential.

A natural candidate criterion to evaluate how good is f_d, is the integral $J(f_d) := \int_{\mathbf{B}} |f - f_d| d\mathbf{x}$, which evaluates the L_1-norm of $f - f_d$ on $\mathbf{B}$, i.e., the "tightness" of $f - f_d$. Indeed, minimizing J tends to minimize the discrepancy (or "error") between f and f_d, uniformly on $\mathbf{B}$. If desired, some flexibility is permitted by allowing any weight function $W : \mathbf{B} \to \mathbb{R}$, positive on $\mathbf{B}$, so as to minimize $J_W = \int_{\mathbf{B}} |f - f_d|\, W d\mathbf{x}$. The above two requirements on f_d as well as the evaluation of $J(f_d)$ are far from being trivial tasks.

Fortunately, to certify $f - f_d \geq 0$ and $\nabla^2 f_d \succeq 0$ on $\mathbf{B}$, a powerful tool is available, namely Putinar's Positivstellensatz [62] (or algebraic positivity certificate), already extensively used in many other contexts, and notably in global polynomial optimization as explained in Chapter 2; see Theorem 2.4.1 and also its many applications in [44]. Moreover, since $f \geq f_d$ on $\mathbf{B}$, the criterion $J(f_d)$ to minimize becomes $\int_{\mathbf{B}} (f - f_d) d\mathbf{x}$ and is *linear* in the coefficients of f_d!

Therefore we end up with a hierarchy of semidefinite programs parametrized by some integer $k \in \mathbb{N}$. This parameter k reflects the size (or complexity) of Putinar's positivity certificates (2.15) for $f - f_d \geq 0$ on $\mathbf{B}$ and for $\nabla^2 f_d \succeq 0$ on $\mathbf{B}$. Any optimal solution of a semidefinite program in this hierarchy provides a convex degree-d polynomial underestimator $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$ with good tightness properties.

We then provide a sequence of convex degree-d polynomial underestimators $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]_d$, $k \in \mathbb{N}$, such that as k increases, $\|f - f_{dk}\|_1 \to \|f - f_d^*\|_1$ for the L_1-norm on $\mathbf{B}$, where f_d^* minimizes $J(h)$ over all convex degree-d polynomial underestimators h of f on $\mathbf{B}$. In fact, any accumulation point φ^* of the sequence $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]_d$ also minimizes $J(h)$ and $f_{dk_i} \to \varphi^*$ pointwise for some subsequence.

This convergence analysis which provides a theoretical justification of the above methodology is only theoretical because in practice one let k fixed (and even to a small value). However, we also prove that if k is sufficiently large, then f_{dk} is necessarily better than $\alpha\mathbf{BB}$-type convex underestimators. A practical validation is also obtained from a comparison with the $\alpha\mathbf{BB}$ method carried out on a set of test examples taken from the literature. Recall that the main motivation for computing underestimators is to compute "good" lower bounds on a box $\mathbf{B}$ for non convex problems, and use these lower bounds in a B&B algorithm. Therefore, to compare the two underestimators we have computed the lower bound obtained my minimizing each of them on a box $\mathbf{B}$. In all examples the lower bounds as well as the tightness score obtained with the moment approach are significantly better.

It is worth emphasizing that our convex underestimators can be also implemented in the improvements of the α**BB** method proposed by Meyer and Floudas [57] or Gounaris and Floudas [22, 23]. Namely, since in those two variants [22, 57] a standard α**BB**-type convex underestimator f_k is computed in each one of the many sub-boxes $\mathbf{B}_k$ of the box $\mathbf{B}$, one may then replace each f_k by a new one $\hat{f}_k$ as proposed in this chapter, and then construct a global one as in [22, 57]. The rationale behind this strategy is that since our estimator $\hat{f}_k$ is tighter than f_k in each box $\mathbf{B}_k$ (at least in all examples that we have tried), the resulting global convex underestimator should also be better in the box $\mathbf{B}$.

In addition, one may also use the moment-SOS approach to provide an alternative way to compute the coefficients α of an α**BB**-type convex underestimator. Namely, since our integral criterion $J(h)$ is a good measure of tightness for any underestimator h, we propose to compute the coefficients α which minimize $\int_{\mathbf{B}} |f - \mathcal{L}| d\lambda$ (where $\mathcal{L}$ is the α**BB**-underestimator), which reduces to solving a *single* semidefinite program! A library of such α's could be computed off-line for several important particular cases.

At last but not least we also provide *parametric* convex underestimators on the box $\mathbf{B}_s := [0, s]^n$ where the scalar s parametrizes the box size. The polynomial underestimator is now a degree-d polynomial in $\mathbb{R}[\mathbf{x}, s]$, which is convex on $\mathbf{x}$ for all values of $s \in [0, S]$ (for some fixed S). This can be especially useful in mixed integer non linear programs where in the Branch & Bound search tree, one has to repeatedly compute lower bounds on boxes of various sizes. Then it may be worth computing *off line* a convex underestimator $f_d \in \mathbb{R}[\mathbf{x}, s]$ and then *on line* in the search tree, one instantiates $\mathbf{x} \mapsto f_d(\mathbf{x}, s) \in \mathbb{R}[\mathbf{x}]_d$ when s is fixed at the value of the desired box size.

3.2 Convex underestimators

3.2.1 Explicit convex underestimators and convex envelopes

There are many results for computing convex underestimators or convex envelopes of simple functions (of a few variables) in explicit form. The basic underlying idea common is to replace a nonconvex term by some new variables that satisfy some constraints to express their link with the non convex term. In most of cases, the relationships are expressed by linear constraints.

For instance, for a bilinear term xy, Al-Khayyal and Falk [5] have showed that the tightest convex lower bound over the domain $[x^L, x^U] \times [y^L, y^U]$ is the piecewise linear function

$$(x, y) \mapsto \quad \max\{x^L y + y^L x - x^L y^L; x^U y + y^U x - x^U y^U\}.$$

Therefore it suffices to introduce

- a new variable ω_B and
- the two linear constraints $\omega_B \geq x^L y + y^L x - x^L y^L$ and $\omega_B \geq x^U y + y^U x - x^U y^U$,

and replace every occurrence of xy with the new variable ω_B.

For the trilinear term xyz, the following eight linear functions in x, y, z whose maximum is a tight convex lower bounding function in the the rectangular region $[x^L, x^U] \times [y^L, y^U] \times [z^L, z^U]$ (see [54]) :

$$\begin{aligned} xyz \geq \max\{&xy^Lz^L + x^Lyz^L + x^Ly^Lz - 2x^Ly^Lz^L,\\ &xy^Uz^U + x^Uyz^L + x^Uy^Lz - x^Uy^Lz^L - x^Uy^Uz^U,\\ &xy^Lz^L + x^Lyz^U + x^Ly^Uz - x^Ly^Uz^U - x^Ly^Lz^L,\\ &xy^Uz^L + x^Uyz^U + x^Ly^Uz - x^Ly^Uz^L - x^Uy^Uz^U,\\ &xy^Lz^U + x^Lyz^L + x^Uy^Lz - x^Uy^Lz^U - x^Ly^Lz^L,\\ &xy^Lz^U + x^Lyz^U + x^Uy^Uz - x^Ly^Lz^U - x^Uy^Uz^U,\\ &xy^Uz^L + x^Uyz^L + x^Ly^Lz - x^Uy^Uz^U - x^Ly^Lz^L,\\ &xy^Uz^U + x^Uyz^U + x^Uy^Uz - 2x^Uy^Uz^U\}. \end{aligned}$$

Moreover, an explicit set of formulas for convex envelopes can be defined for each set of conditions (see [55]).

For the fractional term x/y, Maranas and Floudas (see [54]) have provided a convex underestimator by adding a new variable ω_F and two new linear constraints which depend on the sign of the bounds on x.

$$\omega_F \geq \begin{cases} x^L/y + x/y^U - x^L/y^U & \text{if} \quad x^L \geq 0 \\ x/y^U - x^L y/y^L y^U + x^L/y^L & \text{if} \quad x^L < 0 \end{cases}$$

$$\omega_F \geq \begin{cases} x^L/y + x/y^U - x^L/y^U & \text{if} \quad x^U \geq 0 \\ x/y^U - x^L y/y^L y^U + x^L/y^L & \text{if} \quad x^U < 0. \end{cases}$$

Furthermore, Tawarmalani and Sahinidis, 2001 (see [72]) have provided an explicit form of convex envelopes for x/y in the interval $(x, y) \in [\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}]$ where $\underline{x} \geq 0, \underline{y} \geq 0$.

For the quadrilinear term $xyzt$, Cafieri et al. (see [11]) have provided four types of linear relaxations corresponding to the following grouping of terms $((xy)z)t$, $(xy)(zt)$, $(xyz)t$ and $(xy)zt$. After using the results of convex envelopes for bilinear or trilinear terms, the authors provided a theoretical evidence for what is the best relaxation.

Convex envelopes and convex underestimators for other explicit forms have also been presented. See for instance [49, 50]) for odd degree univariate monomials and [56] for so-called edge-concave functions.

3.2.2 The αBB-type convex underestimators and their refinements

For a general non convex function f a popular approach is the so-called α**BB** method in which the convex underestimator is obtained from the original function f by adding a negative part that satisfy some requirements. This part could be a negative quadratic polynomial of the form

$$\mathbf{x} \mapsto \mathcal{L}(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i (x_i - x_i^L)(x_i - x_i^U), \tag{3.3}$$

as in e.g. Androulakis et al. (see [6]), or an exponential term from the original function of the form

$$\mathbf{x} \mapsto \mathcal{L}(\mathbf{x}) = f(\mathbf{x}) - \sum_{i=1}^{n} (1 - \mathrm{e}^{\alpha_i (x_i - x_i^L)})(1 - \mathrm{e}^{\alpha_i (x_i^U - x_i)}),$$

e.g., as in Akrotirianakis and Floudas (see [4]).

In using such convex underestimators, the only remaining difficulty is to calculate appropriate values for the parameters α_i, $i = 1, \ldots, n$. In doing so one is faced with two conflicting conditions:

- The convexity condition: The negative part must be *large* enough to guarantee convexity of the underestimator $\mathcal{L}(\mathbf{x})$.

- The convergence condition: The negative part also must be as *small* as possible for *tightness* of the underestimator (and hence for the quality of the lower bounds in a B&B algorithm).

In the framework of the α**BB** method several techniques have been proposed for computing the α_i parameters, using interval analysis combined with the convexity conditions. Since $\mathcal{L}$ is convex on $\mathbf{B}$ if and only if its Hessian matrix $\nabla^2\mathcal{L}$ is positive semidefinite on $\mathbf{B}$, the α_i parameters must satisfy

$$\nabla^2\mathcal{L}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) + 2\Delta \succeq 0, \quad \forall \mathbf{x} \in \mathbf{B}, \tag{3.4}$$

where $\Delta := \mathrm{diag}\{\alpha_1, \alpha_2, ..., \alpha_n\}$ is referred to as the *diagonal shift matrix*. For the general case where the Hessian matrix $\nabla^2 f$ is a nonconvex matrix polynomial, computing a good matrix Δ satisfying (3.4) is a difficult task. This task can be alleviated through a transformation of the exact $\mathbf{x}$-dependent Hessian matrix into an interval matrix $\mathcal{H}_f$ such that $\nabla^2 f(\mathbf{x}) \in [\mathcal{H}_f], \forall \mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U]$.

Notice: An interval matrix $[\mathcal{H}_f]$ is a matrix whose elements are interval numbers. An interval matrix $[\mathcal{H}_f]$ will be written as

$$[\mathcal{H}_f] = \tilde{a}_{ij\,(m\times n)} = \begin{vmatrix} \tilde{a}_{11} & ... & \tilde{a}_{1n} \\ ... & ... & ... \\ \tilde{a}_{m1} & ... & \tilde{a}_{mn} \end{vmatrix} \quad \text{where each } \ \tilde{a}_{ij} = [\underline{a_{ij}}, \overline{a_{ij}}].$$

The elements of the original Hessian matrix are treated as independent when calculating their natural interval extensions (see [58, 64]). The interval Hessian matrix family $[\mathcal{H}_f]$ is then used to obtain the following result:

Theorem 3.2.1 (Maranas and Floudas [17, 53])**.** *Let* $[\mathcal{H}_f]$ *be a real symmetric interval matrix such that* $\nabla^2 f(\mathbf{x}) \in [\mathcal{H}_f], \forall \mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U]$*. If* $[\nabla^2_{\mathcal{L}}] := [\mathcal{H}_f] + 2\Delta \succeq 0$ *then* $\mathcal{L}$ *is convex on* $[\mathbf{x}^L, \mathbf{x}^U]$*.*

One of the most efficient method based on interval Hessian matrix is called the *scaled Gershgorin method* in which the $\alpha_i' s$ are determined by

$$\alpha_i = \max\Big\{0, -\frac{1}{2}(\underline{f}_{ii} - \sum_{j\neq i} \max\{|\underline{f}_{ii}|, |\overline{f}_{ij}|\})\frac{d_j}{d_i}\Big\}, \tag{3.5}$$

where $\underline{f}_{ij}$ and $\overline{f}_{ij}$ are lower and upper bounds of $\partial^2 f/\partial x_i \partial x_j$ on the interval $[\mathbf{x}^L, \mathbf{x}^U]$ and $d_i, i = 1, 2, ..., n$ are some chosen positive parameters. For instance in practical, one can chose $d_i = u_i^U - u_i^L$ to reflect the fact that the underestimator is more sensitive to variables with a wide range than to variables with a small range. In fact, regardless of the method being used, the size of the intervals $[x_i^L, x_i^U]$ within the Hessian matrix affects the final accuracy of the computation. Therefore some α**BB** refinements have been introduced to reduce the size of the intervals. See [22, 23, 57] for instance.

However, a crucial remark is to observe that in the basic α**BB** method and its refinements, the *tightness* score $\int_{\mathbf{B}} |f - \mathcal{L}|\, d\mathbf{x}$ of the resulting convex underestimator $\mathcal{L}$ in (3.3) is not taken into account *directly* when one computes the α_i parameters. In the next Section we explain how to build up polynomials convex underestimators that minimize this meaningful tightness score.

3.3 The moment-SOS approach

As already mentioned, a natural candidate criterion to evaluate how *tight* is an underestimator $\mathcal{L}$ of f on a box $\mathbf{B}$, is the integral $J(\mathcal{L}) := \int_{\mathbf{B}} |f - \mathcal{L}|\, d\mathbf{x}$, which evaluates the L_1-norm $\|f - \mathcal{L}\|_1$ of $f - \mathcal{L}$ on $\mathbf{B}$.

Therefore an appropriate model to compute a polynomial convex underestimator of $f \in \mathbb{R}[\mathbf{x}]_d$ is provided by the following optimization problem:

$$\inf_{h \in \mathbb{R}[\mathbf{x}]_d} \left\{ \int_{\mathbf{B}} (f - h) d\lambda \;:\; f - h \geq 0 \text{ on } \mathbf{B};\; h \text{ convex on } \mathbf{B} \right\}, \tag{3.6}$$

where λ is the Borel probability measure uniformly distributed on $\mathbf{B}$. In fact, as we will see later, any finite Borel measure on $\mathbf{B}$ absolutely continuous with respect to λ is fine provided that we can compute all (or sufficiently many of) its moments.

Recall that $f \in \mathbb{R}[\mathbf{x}]_d$ is convex on $\mathbf{B}$ if and only if $\nabla^2 f$ is positive semidefinite on $\mathbf{B}$. Therefore if we define the mapping $\mathbf{T} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}[\mathbf{x}, \mathbf{y}]$ as:

$$h \mapsto \mathbf{T}h(\mathbf{x}, \mathbf{y}) := \mathbf{y}' \nabla^2 h(\mathbf{x}) \mathbf{y}, \quad \forall h \in \mathbb{R}[\mathbf{x}],$$

then f is convex if and only if $\mathbf{T}f_d \geq 0$ on $\mathbf{S} = \mathbf{B} \times \mathbf{U}$, where $\mathbf{U} = \{\mathbf{y} \in \mathbb{R}^n : \sum_{i=1}^n y_i^2 \leq 1\}$.

Moreover, since $\int_{\mathbf{B}} (f - h) d\lambda = \underbrace{\int_{\mathbf{B}} f d\lambda}_{\text{constant}} - \int_{\mathbf{B}} h d\lambda$, the optimization problem (3.6) is equivalent to:

$$\rho_d = \sup_{h \in \mathbb{R}[\mathbf{x}]_d} \left\{ \int_{\mathbf{B}} h d\lambda \;:\; f - h \geq 0 \text{ on } \mathbf{B};\; \mathbf{T}\, h \geq 0 \text{ on } \mathbf{S} \right\}. \tag{3.7}$$

Existence of an optimal solution $f_d^* \in \mathbb{R}[\mathbf{x}]_d$ of problem (3.7) is proved in *Lemma 3.1* (of the paper "Convex underestimators of polynomials" in Appendix A). In addition, as a direct consequence of Theorem 2.4.1, one may replace the constraint "$f - h \geq 0$ on $\mathbf{B}$" (resp. "$\mathbf{T}\,h \geq 0$ on $\mathbf{S}$") with "$f - h \in Q_{\mathbf{B}}$" (resp. "$f - h \in Q_{\mathbf{S}}$") where $Q_{\mathbf{B}} \subset \mathbb{R}[\mathbf{x}]$ (resp. $Q_{\mathbf{S}} \subset \mathbb{R}[\mathbf{x}, \mathbf{y}]$) is the quadratic module[1] associated with $\mathbf{B}$ (resp. $\mathbf{S}$). That is, an optimal solution f_d^* of (3.6) is an optimal solution of the problem $\mathbf{P}_d$ defined by:

$$\mathbf{P}_d : \qquad \rho_d = \sup_{h \in \mathbb{R}[\mathbf{x}]_d} \left\{ \int_{\mathbf{B}} h \, d\lambda \;:\; f - h \in Q_{\mathbf{B}};\; \mathbf{T}h \in Q_{\mathbf{S}} \right\}. \tag{3.8}$$

3.3.1 Semidefinite relaxations for computing a convex underestimator

Given the box $\mathbf{B} = [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n$, let g_0 be the constant polynomial $g_0 = 1$, and define the quadratic polynomials $\mathbf{x} \mapsto g_j(\mathbf{x}) = (x_i^U - x_j)(x_j - x_j^L)$, $j = 1, \ldots, n$, as well as $\mathbf{x} \mapsto g_{n+1}(\mathbf{x}) = 1 - \|\mathbf{x}\|^2$.

Associated with the optimization problem $(\mathbf{P}_d)$, we can construct a hierarchy of semidefinite relaxations $(\mathbf{P}_{dk})$ of problem $\mathbf{P}_d$ as follows:

$$\left\{ \begin{array}{ll} \displaystyle\sup_{h \in \mathbb{R}[\mathbf{x}]_d, \sigma_j, \theta_\ell} & \int_{\mathbf{B}} h \, d\lambda \\ & f(\mathbf{x}) = h(\mathbf{x}) + \displaystyle\sum_{j=0}^{n} \sigma_j(\mathbf{x}) g_j(\mathbf{x}) \quad \forall \mathbf{x} \\ \text{s.t.} & \mathbf{T}h(\mathbf{x}, \mathbf{y}) = \displaystyle\sum_{j=0}^{n} \theta_j(\mathbf{x}, \mathbf{y}) g_j(\mathbf{x}) \\ & \qquad + \theta_{n+1}(\mathbf{x}, \mathbf{y}) g_{n+1}(\mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y} \\ & \sigma_0 \in \Sigma[\mathbf{x}]_k,\ \sigma_j \in \Sigma[\mathbf{x}]_{k-1},\ j \geq 1 \\ & \theta_0 \in \Sigma[\mathbf{x}, \mathbf{y}]_k, \theta_j \in \Sigma[\mathbf{x}, \mathbf{y}]_{k-1},\ j \geq 1, \end{array} \right. \tag{3.9}$$

where $k \geq \max[\lceil d/2 \rceil, \lceil (\deg f)/2 \rceil]$ and with optimal value denoted by ρ_{dk}.

Problem (3.9) is an obvious relaxation of problem $\mathbf{P}_d$ in (3.7). It is a semidefinite program whose dual is the semidefinite program:

[1] Recall that if $\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n \;:\; g_j(\mathbf{x}) \geq 0,\ j = 1, \ldots, m\}$ for some polynomials $(g_j) \subset \mathbb{R}[\mathbf{x}]$, the *quadratic module* $Q_{\mathbf{K}} \subset \mathbb{R}[\mathbf{x}]$ associated with $\mathbf{K}$ is defined by $Q_{\mathbf{K}} = \{\sum_{j=0}^{m} \sigma_j \, g_j \;:\; \sigma_j \in \Sigma[\mathbf{x}]\}$, where g_0 is the constant polynomial $g_0(\mathbf{x}) = 1$, and $\Sigma[\mathbf{x}] \subset \mathbb{R}[\mathbf{x}]$ is the space of SOS polynomials.

$$\begin{cases} \inf\limits_{\mathbf{z},\mathbf{v}} & L_{\mathbf{v}}(f) \\ \text{s.t.} & \mathbf{M}_k(g_j\mathbf{v}) \succeq 0, \qquad j=0,...,n \\ & \mathbf{M}_k(g_{n+1}\,\mathbf{z}) \succeq 0 \\ & \mathbf{M}_k(g_j\,\mathbf{z}) \succeq 0, \qquad j=0,...,n \\ & v_\alpha - (\mathbf{T}^*\mathbf{z})_\alpha = \gamma_\alpha, \qquad \alpha \in \mathbb{N}^n_d. \end{cases} \tag{3.10}$$

where $\gamma_\alpha = \int_{\mathbf{B}} \mathbf{x}^\alpha d\lambda, \mathbf{v} = (v_\alpha)\,\forall\alpha \in \mathbb{N}^n_{2k}$ and $\mathbf{z} = (z_{\alpha\beta})\ \forall\alpha,\beta \in \mathbb{N}^n$ with $|\alpha+\beta| \leq 2k$. The optimal value of $\mathbf{P}^*_{dk}$ is denoted by ρ^*_{dk}.

Theorem 3.3.1. *With $d \in \mathbb{N}$ fixed, consider problem $\mathbf{P}_d$ in (3.7) with optimal value ρ_d and the associated semidefinite relaxations (3.9).*

(i) *The semidefinite program (3.9) has an optimal solution $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$ when $2k$ is sufficiently large and satisfies $2k \geq degf$.*

(ii) *The sequence of optimal values (ρ_{dk}), $k \in \mathbb{N}$, associated with the hierarchy of semidefinite relaxations (3.9) is monotone non decreasing, and $\int_{\mathbf{B}} f d\lambda - \rho_{dk} \to \rho_d$ as $k \to \infty$.*

(iii) *In addition, any accumulation point $\varphi^* \in \mathbb{R}[\mathbf{x}]_d$ of the sequence $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]_d$, is an optimal solution of (3.7), and $f_{dk_i} \to \varphi^*$ pointwise for some subsequence (k_i), $i \in \mathbb{N}$.*

The proof is given in the paper details.

Even though we can approximate the optimal value of problem (3.7) as closely as desired by solving the hierarchy of semidefinite programs (3.9), the size of the moment and localizing matrices $\mathbf{M}_k(g_j\,\mathbf{v})$ and $\mathbf{M}_k(g_j\,\mathbf{z})$ increases fast with k. Therefore in practice one let k *fixed* at a small value. However, the computational experiments presented below indicate that even with k fixed at its smallest possible value, the resulting polynomial underestimator f_{dk} provides (significantly) better lower bounds than the α**BB**-underestimator.

3.3.2 Computing α in αBB-type convex underestimators

In fact we can also use the above approach to provide a new and systematic way to compute the coefficients α_i, $i:1,\dots,n$, of an α**BB**-type convex underestimator $\mathcal{L}$ of the form (3.3). That is, one may define a procedure to compute the α_i-parameters which minimize the tightness criterion $\int_{\mathbf{B}} |f-\mathcal{L}|d\mathbf{x}$. Recall that in the various α**BB** methods the tightness criterion is not taken into account *directly* as we do here.

Indeed, after a rescaling of the box $\mathbf{B} = [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n$ to $[0,1]^n$, and in view of the definition (3.3) of $\mathcal{L}$, one wishes to minimize:

$$\begin{aligned} J(\mathcal{L}) \;=\; \int_{\mathbf{B}} |f - \mathcal{L}|\, d\lambda \;=\; \int_{\mathbf{B}} (f - \mathcal{L})\, d\lambda \;&=\; \underbrace{\int_{\mathbf{B}} f\, d\lambda}_{\text{constant}} + \sum_{i=1}^{n} \alpha_i \int_{\mathbf{B}} x_i(1-x_i)\, d\lambda \\ &=\; \int_{\mathbf{B}} f\, d\lambda + \frac{1}{6} \sum_{i=1}^{n} \alpha_i, \end{aligned} \qquad (3.11)$$

over all convex underestimators $\mathcal{L}$ of the form (3.3). In other words one wants to solve

$$J^* \;:=\; \inf_{\alpha} \{ J(\mathcal{L}) \;:\; \mathcal{L} \text{ as in (3.3) and convex on } \mathbf{B} \}. \qquad (3.12)$$

As already done in the previous section, the convexity constraint can be replaced with Putinar's positivity certificate $\mathbf{y}'\nabla^2\mathcal{L}(\mathbf{x})\mathbf{y} \in Q_{\mathbf{S}}$, or equivalantly,

$$\mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y} + 2\sum_{i=1}^{n} \alpha_i\, y_i^2 \;\in\; Q_{\mathbf{S}},$$

so that (3.12) is equivalent to:

$$\eta \;=\; \inf_{\alpha} \left\{ \sum_{i=1}^{n} \alpha_i \;:\; \mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y} + 2\sum_{i=1}^{n} \alpha_i\, y_i^2 \;\in\; Q_{\mathbf{S}} \right\}. \qquad (3.13)$$

And as before, an obvious way to approximate η is to replace the constraint

$$\mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y} + 2\sum_{i=1}^{n} \alpha_i\, y_i^2 \;\in\; Q_{\mathbf{S}}$$

with the more tractable constraint

$$\mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y} \;=\; -2\sum_{i=1}^{n} \alpha_i y_i^2 + \sum_{j=0}^{n+1} \theta_j\, g_j\,; \quad \theta_j \in \Sigma[\mathbf{x},\mathbf{y}]_{k-v_j},\; j = 0,\ldots,n+1, \qquad (3.14)$$

where $k \geq \lceil (\deg f)/2 \rceil$, and $v_j = \lceil (\deg g_j)/2 \rceil$, $j = 0,\ldots,n{+}1$, and then let k increase.

Therefore to approximate the α_i's that minimize (3.11) one may solve the optimization problem

$$\begin{aligned} \eta_k = \inf_{\alpha,\theta_j} \quad & \sum_{i=1}^{n} \alpha_i \\ \text{s.t.} \quad & \mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y} = -2\sum_{i=1}^{n} \alpha_i y_i^2 + \sum_{j=0}^{n+1} \theta_j g_j \\ & \alpha \geq 0;\ \theta_j \in \Sigma[\mathbf{x},\mathbf{y}]_{k-v_j},\ j = 0,\ldots,n+1, \end{aligned} \tag{3.15}$$

which is a semidefinite program whose size depends on the size of Putinar's convexity certificate (3.14) (parametrized by k). Once again, the optimal value η in (3.13) can be approximated as closely as desired by solving the hierarchy of semidefinite programs (3.15) whose associated monotone non increasing sequence of optimal values (η_k), $k \in \mathbb{N}$, converges to η as $k \to \infty$.

3.3.3 Parametric convex underestimators

In this section, we build up a family of convex underestimators f^s, $0 \leq s \leq S$, on the box $\mathbf{B}_s := [0, s]^n$ whose size is parametrized by the scalar $s \in [0, S]$ for some S. We are motivated by the fact that in deterministic methods such as *Branch and Bound* (B&B), one has to repeatedly compute online convex under estimators on such boxes. Therefore instead of computing a convex underestimator f^s for every s in some interval say $[0, S]$, we might try to compute in one single shot (or once and for all $s \in [0, S]$), a parametrized family of best degree-d convex polynomial underestimators $(f_d^s) \subset \mathbb{R}[\mathbf{x}]_d$ of f on $\mathbf{B}_s$, $s \in [0, S]$.

Moreover, the "tightness" criterion $\|f - f^s\|_1$ being a good indicator of the quality of the convex underestimator f^s on the box $[0, s]^n$, and in light of parameter optimization, a natural criterion to evaluate the efficiency of a parametrized convex underestimator f^s is the integral

$$\int_0^S \|f - f^s\|_1 \, ds \;=\; \int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - f^s(\mathbf{x})) \, d\mathbf{x} \, ds.$$

For the dependence of f^s on the parameter s, a natural choice is to consider f^s as a degree-d polynomial in s and $\mathbf{x}$, i.e., $f_d^s \in \mathbb{R}[\mathbf{x}, s]_d$, optimal solution of the optimization problem:

$$\left\{\begin{array}{rl} \rho_d = \displaystyle\min_{h\in\mathbb{R}[\mathbf{x},s]_d} & \displaystyle\int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - h(\mathbf{x},s))\, d\mathbf{x} ds \\ & \\ \text{s.t.} & f - h \geq 0 \text{ on } \mathbf{B}_s \times [0,S] \\ & \mathbf{T}\, h \geq 0 \text{ on } \mathbf{B}_s \times [0,S] \times \mathbf{U} \end{array}\right. \tag{3.16}$$

where the linear mapping $\mathbf{T} : \mathbb{R}[\mathbf{x},s] \to \mathbb{R}[\mathbf{x},s,\mathbf{y}]$ is now given by:

$$h \;\mapsto\; \mathbf{T}h(\mathbf{x},s,\mathbf{y}) \;=\; \mathbf{y}'\,\nabla^2_{\mathbf{x}} h(\mathbf{x},s)\,\mathbf{y}, \qquad h \in \mathbb{R}[\mathbf{x},s],$$

and $\nabla^2_{\mathbf{x}} h(\mathbf{x},s) = (\partial^2 h(\mathbf{x},s)/\partial x_i \partial x_j)$, $i,j = 1,\ldots,n$.

For every $s \in [0,S]$, let $f_d^s \in \mathbb{R}[\mathbf{x}]$ be an optimal solution of (3.6) with $\mathbf{B}_s$ in lieu of $\mathbf{B}$, and consider the mapping $s \mapsto \rho_d(s) = \int_{\mathbf{B}_s} f_d^s(\mathbf{x}) d\mathbf{x}$. With $f_d^* \in \mathbb{R}[\mathbf{x},s]$ is an optimal solution of problem (3.16), we have:

$$\begin{aligned} \int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - f_d^*(\mathbf{x},s))\, d\mathbf{x}\, ds \;&\geq\; \rho^* \;:=\; \int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - f_d^s(\mathbf{x}))\, d\mathbf{x}\, ds \\ &=\; \int_\Delta f(\mathbf{x}) d\mathbf{x} ds - \int_0^S \rho_d(s)\, ds. \end{aligned}$$

Ideally one would like to approximate ρ^*. This is possible if one consider polynomials $h(\mathbf{x},s)$ with degree d in $\mathbf{x}$ *and* degree in s arbitrary large. Indeed, if $\rho_d(s)$ is Riemann integrable on $[0,S]$ then

$$\int_0^s \rho_d(s)\, ds \;=\; \lim_{N\to\infty} \frac{S}{N} \sum_{k=1}^N \rho_d(kS/N),$$

and one may find a polynomial $h_N \in \mathbb{R}[\mathbf{x},s]$ of degree d in $\mathbf{x}$ and degree at most N in s, such that $h_N(\mathbf{x},s) = f_d^s(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{B}_s$ and all $s = kS/N$, $k = 0,1,\ldots,N$. Write $f_d^s(\mathbf{x}) = \sum_{\alpha\in\mathbb{N}^n_d} f^s_\alpha \mathbf{x}^\alpha$, for some coefficients (f^s_α), and for every $\alpha \in \mathbb{N}^n_d$ let $p_\alpha \in \mathbb{R}[s]_N$ be such that

$$p_\alpha(kS/N) \;=\; f_\alpha^{kS/N}, \quad k = 0,1,\ldots,N.$$

Then the polynomial

$$(\mathbf{x},s) \mapsto h_N(\mathbf{x},s) \;:=\; \sum_{\alpha\in\mathbb{N}^n_d} p_\alpha(s)\, \mathbf{x}^\alpha,$$

has degree N in s and degree d in $\mathbf{x}$, and matches f_d^s on $\mathbf{B}_{kS/N} \times \{kS/N\}$, $k = 0,\ldots,N$.

Semidefinite approximations

For practical purpose, $h \in \mathbb{R}[\mathbf{x}, s]_{2d}$ in (3.16) is a polynomial of degree d in $\mathbf{x}$ and each coefficient of $\mathbf{x}^\alpha$ is a polynomial of degree d in the parameter s, i.e.,

$$h(\mathbf{x}, s) \;=\; \sum_{\alpha \in \mathbb{N}^n_d} h_\alpha(s)\, \mathbf{x}^\alpha, \qquad h_\alpha \in \mathbb{R}[s]_d. \tag{3.17}$$

And again, one approximates $f^*_{2d} \in \mathbb{R}[\mathbf{x}, s]_{2d}$ by a sequence of polynomials $(f_{dk}) \subset \mathbb{R}[\mathbf{x}, s]_{2d}$, $k \in \mathbb{N}$, of the form (3.17), by solving a hierarchy of semidefinite programs. Let $\hat{g}_j \in \mathbb{R}[\mathbf{x}, s]$, $j = 1, \ldots, n+2$ be the polynomials

$$s \mapsto \hat{g}_j(\mathbf{x}, s) := x_j(s - x_j),\; j = 1, \ldots, n,$$

and

$$\hat{g}_{n+1}(\mathbf{x}, s) := s(S - s); \quad \hat{g}_{n+2}(\mathbf{y}) := 1 - \|\mathbf{y}\|^2.$$

Consider the hierarchy of semidefinite programs:

$$\begin{array}{rl} \displaystyle\max_{h \in \mathbb{R}[\mathbf{x},s]_d, \sigma_j, \theta_k} & \displaystyle\int_0^S \int_{\mathbf{B}_s} h(\mathbf{x}, s)\, d\mathbf{x}\, ds \\ \text{s.t.} & \displaystyle f(\mathbf{x}) = h(\mathbf{x}, s) + \sum_{j=0}^{n+1} \sigma_j(\mathbf{x}, s)\hat{g}_j(\mathbf{x}, s)) \quad \forall \mathbf{x} \\ & \displaystyle \mathbf{T}h(\mathbf{x}, \mathbf{y}) = \sum_{j=0}^{n+1} \theta_j(\mathbf{x}, s, \mathbf{y})\hat{g}_j(\mathbf{x}, s) \\ & \qquad\qquad + \theta_{n+1}(\mathbf{x}, s, \mathbf{y})\hat{g}_{n+2}(\mathbf{y}) \quad \forall \mathbf{x}, s, \mathbf{y} \\ & \\ & \sigma_0 \in \Sigma[\mathbf{x}, s]_k,\; \sigma_j \in \Sigma[\mathbf{x}, s]_{k-1},\; j \geq 1 \\ & \\ & \theta_0 \in \Sigma[\mathbf{x}, s, \mathbf{y}]_k, \theta_j \in \Sigma[\mathbf{x}, s, \mathbf{y}]_{k-1},\; j \geq 1, \end{array} \tag{3.18}$$

with $k \geq \max[\lceil d/2 \rceil, \lceil (\deg f)/2 \rceil]$, and where h is of the form (3.17).

Moreover, we can also consider more general boxes like e.g. two-parameter boxes of the form $\mathbf{B}_s := [s_1, s_2]^n$ with $0 \leq s_1 \leq s_2 \leq S$ for some fixed $S > 0$. But solving the semidefinite program (3.16) is computationally more expensive as the criterion now reads

$$\max_{h \in \mathbb{R}[\mathbf{x}, s_1, s_2]_{2d}, \sigma_j, \theta_k} \int_0^S \int_{s_1}^S \left(\int_{\mathbf{B}_s} h(\mathbf{x}, s_1, s_2)\, d\mathbf{x} \right) ds_2 ds_1,$$

and to define the constraints of (3.18) we also need introduce the polynomials

$$\hat{g}_j(\mathbf{x}, s_1, s_2) = (x_j - s_1)(s_2 - x_j), \quad j = 1, \ldots, n,$$

as well as $\hat{g}_{n+1}(\mathbf{x}, s_1, s_2) = s_1(S - s_1)$, $\hat{g}_{n+2}(\mathbf{x}, s_1, s_2) := (s_2 - s_1)(S - s_2)$, and $\hat{g}_{n+3}(\mathbf{x}, s_1, s_2, \mathbf{y}) = 1 - \|\mathbf{y}\|^2$.

3.4 Comparison and computational results

We have compared our polynomial convex underestimator with α**BB**-type underestimators on test examples of nonconvex optimization polynomials taken from the literature. Among all possible choices of d (the degree of our polynomial convex underestimator f_d), we have considered the two practical choices $d = 2$ and $d =$ deg f . With the first choice, we try to construct a quadratic underestimator of f and with the second one, we search for the best convex underestimator of same degree as f. The results were presented in Table 3.1 and Table 3.2 .

Prob	n	deg f	d	$[\mathbf{x}^L, \mathbf{x}^U]$	$f^*_{\alpha\mathbf{BB}}$	f^*_{mom}	f^*
Test2	4	3	2	[0,1]	-1.54	-1.22	-1
Test3	5	4	2	[-1,1]	-15	-13.95	-6
Test4	6	6	2	[-1,1]	-60.15	-10.06	-3
Test5	3	6	2	[-2,2]	-411.2	-12.66	-1
Test10	4	4	2	[0,1]	-197.54	-0.9698	0
Test11	4	4	2	[0,1]	-33.02	-0.623	0
Test14(1)	3	4	2	[-5,2]	-2409	-300	-300
Test14(2)	4	4	2	[-5,2]	-3212	-400	-400
Test14(3)	5	4	2	[-5,2]	-4015	-500	-500
Fl.2.2	5	2	2	[0,1]	-18.9	-18.9	-17
Fl.2.3	6	2	2	[0,1]	-5270.9	-361.50	-361
Fl.2.4	13	2	2	[0,3]	-592	-195	-195
Fl.2.6	10	2	2	[0,1]	-269.83	-269.45	-268.01
Fl.2.8C1	20	2	2	[0,3]	-560	-560	-394.75
Fl.2.8C2	20	2	2	[0,10]	-1050	-1050	-884
Fl.2.8C3	20	2	2	[0,30]	-13600	-12000	-8695
Fl.2.8C4	20	2	2	[0,30]	-920	-920	-754.75
Fl.2.8C5	20	2	2	[0,30]	-16645	-10010	-4150.41

Table 3.1: Comparing f^*_{mom} and $f^*_{\alpha\mathbf{BB}}$; $d = 2$.

Prob	n	$\deg f$	d	$[\mathbf{x}^L,\mathbf{x}^U]$	$f^*_{\alpha\mathbf{BB}}$	f^*_{mom}	f^*
Test2	4	3	3	[0,1]	-1.54	-1.22	-1
Test3	5	4	4	[-1,1]	-15	-11.95	-6
Test5	3	6	6	[-2,2]	-411.2	-12.07	-1
Test10	4	4	4	[0,1]	-197.54	-0.778	0
Test11	4	4	4	[0,1]	-33.02	0	0
Test14(1)	3	4	4	[-5,2]	-2409	-300	-300
Test14(2)	4	4	4	[-5,2]	-3212	-400	-400
Test14(3)	5	4	4	[-5,2]	-4015	-500	-500

Table 3.2: Comparing f^*_{mom} and $f^*_{\alpha\mathbf{BB}}$; $d = \deg f$.

Recall that one important motivation for computing underestimators is to obtain “good" lower bounds on a box $\mathbf{B}$ for non convex problems, and e.g., use these lower bounds in a B&B algorithm. Therefore, to compare the moment and $\alpha\mathbf{BB}$-type convex underestimators, we have chosen non convex optimization problems from the literature, and replaced the original non convex objective function by our underestimator f_d and an $\alpha\mathbf{BB}$-type convex underestimator $\mathcal{L}$. We then compare the minimum f^*_{mom} (resp. $f^*_{\alpha\mathbf{BB}}$) obtained by minimizing f_d (resp. $\mathcal{L}$) on the box $\mathbf{B}$.

We have also compared the respective tightness scores $\int_{\mathbf{B}} |f - f_d|\, d\mathbf{x}$ and$\int_{\mathbf{B}} |f - \mathcal{L}|\, d\mathbf{x}$ (see Table 3.3).

We have also compared the standard $\alpha\mathbf{BB}$ underestimator with the new $\alpha\mathbf{BB}$ underestimator described in Section §3.3.2 where the α_i's minimize the tightness criterion (see Table 3.4).

Moreover, we also illustrate the application of the moment-SOS method to refinements of $\alpha\mathbf{BB}$-type convex underestimator such as the recent $\alpha\mathbf{BB}$-spline and piecewise-$\alpha\mathbf{BB}$ underestimators. In such refinements the original domain $\mathbf{B}$ is divided into some subdomains $\mathbf{B}_i$, $i \in \mathbf{I}$, and a global underestimator is obtained by combining in certain manner local underestimators computed in each domain $\mathbf{B}_i$, $i \in \mathbf{I}$. Therefore we propose to compute local polynomial underestimators in each subdomain $\mathbf{B}_i$ via the moment-SOS method described earlier instead of the Scaled Gershgorin method for $\alpha\mathbf{BB}$ underestimators. In our (limited) computational experiments (showed in Table 3.5) we have also obtained significantly better results.

All computations were made by running the Gloptipoly software described in Henrion et al., and developed for solving the Generalized Problem of Moments whose

Prob	$\frac{\int_{\mathbf{B}}\|f-\mathcal{L}\|d\lambda}{V_{\mathbf{B}}}$	$\frac{\int_{\mathbf{B}}\|f-f_d\|d\lambda}{V_{\mathbf{B}}}$
Test2	1	0.625
Test3	11.67	3.33
Test4	60	7.29
Test5	99.00	23.20
Test10	133.33	57.00
Test11	46.33	1
Test14(1)	1641.4e+003	149.2711
Test14(2)	2186.6	199.08
Test14(3)	2731	248.71
Fl.2.2	41.66	41.66
Fl.2.3	67500	833.33
Fl.2.4	0.005	0.0005
Fl.2.6	8.33	5.83
Fl.2.8C1	3.4989e+014	3.4989e+014
Fl.2.8C2	12200	12200
Fl.2.8C3	6.9979e-005	6.9979e-005
Fl.2.8C4	3.4989e-006	3.4989e-006
Fl.2.8C5	9.6077e-015	9.6077e-015

Table 3.3: Comparing $\frac{\int_{\mathbf{B}}|f-\mathcal{L}|d\lambda}{V_B}$ and $\frac{\int_{\mathbf{B}}|f-f_d|d\lambda}{V_B}$

Prob	$[x^L, x^U]$	$f^*_{\alpha\mathbf{BB}}$	f^*_{mom}	f^*
Fl 8.2.7	$[0,1]^5$	-899.5	-2.76	-0.5
Fl 8.2.7	$[-1,1]^5$	-2999	-23	-0.6
Fl 8.2.7	$[-5,5]^5$	-63000	-2987	-982
Test 10	$[0,1]^4$	-197.5	-61.9	0
Test 10	$[-1,1]^4$	-870.2	-323.8	0
Test 10	$[-5,5]^4$	-137e+05	-4.73e+04	-19

Table 3.4: Comparing the $f_{\alpha\mathbf{BB}}$ and f_d^α underestimators.

global optimization is only a special case. In all examples, the α**BB**-type convex underestimators were computed via the Scaled Gershgorin method.

Prob	n	$\deg f$	d	$[\mathbf{x}^L,\mathbf{x}^U]$	f_1^*	f_2^*	f_3^*	f_4^*	f^*
Test2	4	3	3	[0,1]	-1.54	-1.006	-1.001	-1.22	-1
Test3	5	4	3	[-1,1]	-15	-6.07	-6	-13.95	-6
Test4	6	6	2	[-1,1]	-60.15	-18.5	-4.05	-10.06	-3
Test5	3	6	3	[-2,2]	-411.2	-37.79	-12.17	-12.66	-1
Test10	4	4	3	[0,1]	-197.54	-14.78	-0.8	-0.9698	0
Test11	4	4	3	[0,1]	-33.02	-19.77	-0.007	-0.623	0
Test14(1)	3	4	3	[-5,2]	-2409	-855	-300	-300	-300
Test14(2)	4	4	3	[-5,2]	-3212	-1141.2	-400	-400	-400
Test14(3)	5	4	3	[-5,2]	-4015	-1426.5	-500	-500	-500
Fl.2.2	5	2	3	[0,1]	-18.9	-18.77	-17	-18.9	-17
Fl.2.3	6	2	2	[0,1]	-5270.9	-2176.4	-361	-361.50	-361

Table 3.5: d:number of interval in each coordinate, $f_1^*{=}f^*_{\alpha\mathbf{BB}}$, $f_2^*{=}f^*_{spline-\alpha\mathbf{BB}}$, $f_3^*{=}f^*_{spline-mom}$, $f_4^*{=}f^*_{mom}$

3.5 Conclusion

In this chapter we have presented a new application of moment-SOS approach to construct convex polynomial underestimators for a nonconvex polynomial f on a box $\mathbf{B} \subset \mathbb{R}^n$. Ideally this polynomial underestimator $\tilde{f}$ should minimize the important tightness criterion $\rho = \int_{\mathbf{B}} |f - \tilde{f}|\, d\mathbf{x}$. We have shown how to approximate ρ as closely as desired by solving a hierarchy of semidefinite programs.

In fact, for practical purpose and computational efficiency we only use the convex underestimator obtained at the first step in the hierarchy. However and perhaps surprisingly, even though the resulting convex underestimator is obtained at the first step in the hierarchy, it performs significantly better than a “standard" $\alpha\mathbf{BB}$-type underestimator, at least on a sample of non convex problems from the literature. Not only its tightness score is better but also the lower bound obtained by minimizing this convex underestimator is also significantly better than the one obtained by minimizing the popular $\alpha\mathbf{BB}$-type underestimator. In addition, the moment-SOS approach can also be applied

- to improve $\alpha\mathbf{BB}$-type underestimators by computing the α_i's that minimize the tightness criterion.

- to replace “local" $\alpha\mathbf{BB}$-type underestimators in some refinements of the $\alpha\mathbf{BB}$ method like in [22, 23, 57].

In both cases numerical experiments on a (limited) sample of problems have demonstrated siginificants improvements

Finally, we have also provided parametric convex polynomial underestimators

$h \in \mathbb{R}[\mathbf{x}, s]$ where the parameter $s \in [0, S]$ defines the size of the box $[0, s]^n$ where one wishes to underestimate f by a convex polynomial $f^s \in \mathbb{R}[\mathbf{x}]$, namely $\mathbf{x} \mapsto f^s(\mathbf{x}) = h(\mathbf{x}, s)$ with s fixed. And so this family of convex underestimators (f^s), $0 \leq s \leq S$, can be computed off-line, which permits to avoid computing on-line at each node of the search tree of a B& B method, a convex underestimator on $[0, s]^n$, for each desired value of $s \in [0, S]$.

3.6 The paper details

See Appendix A: *Convex underestimators of polynomials* appeared in Journal of Global Optimization, (2012) pp 1-25, doi:10.1007/s10898-012-9974-4.

Chapter 4

A "Joint+marginal" algorithm for optimization

4.1 Introduction

Consider the general polynomial program:

$$\mathbf{P}: \quad f^* = \min_{\mathbf{x}} \{ f(\mathbf{x}) \, : \, g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, m \} \tag{4.1}$$

where $f, g_j \in \mathbb{R}[\mathbf{x}]$, $j = 1, \ldots, m$. Let $\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n \, : \, g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, m\} \subset \mathbb{R}^n$ be the feasible set and f^* be the global minimum of $\mathbf{P}$.

One may use the moment-SOS approach described in Chapter 1 to approximate f^* as closely as desired and sometimes to obtain f^* exactly as well as some global minimizer(s) $\mathbf{x}^* \in \mathbf{K}$. However, to deal with large size and even medium size problems $\mathbf{P}$, only a few relaxations of the hierarchy of semidefinite relaxations (2.16) can be implemented (the first, second or the third relaxation) and so one only obtain a lower bound ρ_d on f^* and no feasible solution $\mathbf{x} \in \mathbf{K}$ in general.

So a natural issue to consider is the following: *"When only a few relaxations can be solved, how can we use such relaxations (or a slight modification) to help find a feasible solution of the original problem* $\mathbf{P}$*?*

In Chapter 3 we have seen how to use the moment-SOS approach to provide convex underestimators for f (and for $-g_j$ when g_j is not concave) on a box $\mathbf{B} \supset \mathbf{K}$. And for instance, such convex under estimators can be used for computing good lower bounds in a Branch & Bound procedure to approximate f^*. In this chapter we take a different approach and provide another application of the moment-SOS approach. It uses a simple idea combined with the so-called "Joint+Marginal" approach (in short (J+M)-approach) developed in [45] for parametric polynomial optimization and results in a relatively simple *heuristic* method for the general polynomial problem **P**.

Brief outline

With problem **P** as in (4.1) and $k = 1, \dots, n$, let the compact interval $\mathbf{Y}_k := [\underline{x}_k, \overline{x}_k] \subset \mathbb{R}$ be contained in the projection of **K** into the x_k-coordinate axis. In the context of the (non-parametric) polynomial optimization **P**, the (J+M)-approach can be used as follows in what we call the **(J+M)-algorithm**:

• (a) Treat x_1 as a parameter in the compact interval $\mathbf{Y}_1 = [\underline{x}_1, \overline{x}_1]$ with associated probability distribution φ_1 uniformly distributed on $\mathbf{Y}_1$.

• (b) Construct a hierarchy of semidefinite relaxations (in spirit of moment-SOS approach) for the problem $\mathbf{P}(x_1)$ with $n-1$ variables $(x_2, \dots, x_n)$ and parameter x_1, which is problem **P** with the additional constraint that the variable $x_1 \in \mathbf{Y}_1$ is fixed. With $i \in \mathbb{N}$ **fixed**, solve the i-th semidefinite relaxation; an optimal solution of its dual provides a univariate polynomial $x_1 \mapsto J_i^1(x_1)$ which, if i would increase, would converge to $J^1(x_1)$ in the $L_1(\varphi_1)$-norm. (The map $v \mapsto J^1(v)$ denotes the optimal value function of $\mathbf{P}(v)$, i.e. the optimal value of **P** given that the variable x_1 is fixed at the value v).

• (c) Next, compute $\tilde{x}_1 \in \mathbf{Y}_1$, a global minimizer of the univariate polynomial J_i^1 on $\mathbf{Y}_1$ (e.g. this can be done by solving a single semidefinite program). Ideally, when i is large enough, $\tilde{x}_1$ should be close to the first coordinate x_1^* of a global minimizer $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$ of **P**.

• (d) Go back from step (b) with now $x_2 \in \mathbf{Y}_2 \subset \mathbb{R}$ instead of x_1, and with φ_2 being the probability measure uniformly distributed on $\mathbf{Y}_2$. With the same method, compute a global minimizer $\tilde{x}_2 \in \mathbf{Y}_2$, of the univariate polynomial $x_2 \mapsto J_i^2(x_2)$ on the interval $\mathbf{Y}_2$. Again, if i would increase, J_i^2 would converge in the $L_1(\varphi_2)$-norm to the optimal value function $v \mapsto J^2(v)$ of $\mathbf{P}(x_2)$ (i.e. the optimal value of **P** given that the variable x_2 is fixed at the value v.) Iterate until one has obtained $\tilde{x}_n \in \mathbf{Y}_n \subset \mathbb{R}$.

One ends up with a point $\tilde{\mathbf{x}} \in \prod_{k=1}^n \mathbf{Y}_k$ and in general $\tilde{\mathbf{x}} \notin \mathbf{K}$. This point $\tilde{\mathbf{x}}$ can be used as initial guess of a local optimization procedure to find a local minimum $\hat{\mathbf{x}} \in \mathbf{K}$. The rationale behind the (J+M)-algorithm is that if i is large enough and **P** has a unique global minimizer $\mathbf{x}^* \in \mathbf{K}$, then $\tilde{\mathbf{x}}$ as well as $\hat{\mathbf{x}}$ should be close to $\mathbf{x}^*$.

When the feasible set $\mathbf{K}$ is convex, one may define the following variant to obtain a *feasible* point $\tilde{\mathbf{x}} \in \mathbf{K}$. Again, let $\mathbf{Y}_1$ be the projection of $\mathbf{K}_1$ into the x_1-coordinate axis. Once $\tilde{x}_1 \in \mathbf{Y}_1$ is obtained in step (b), consider the new optimization problem $\mathbf{P}(\tilde{x}_1)$ in the $n-1$ variables $(x_2, \ldots, x_n)$, obtained from $\mathbf{P}$ by fixing the variable $x_1 \in \mathbf{Y}_1$ at the value $\tilde{x}_1$. Its feasible set is the convex set $\mathbf{K}_1 := \mathbf{K} \cap \{\mathbf{x} : x_1 = \tilde{x}_1\}$. Let $\mathbf{Y}_2$ be the projection of $\mathbf{K}_1$ into the x_2-coordinate axis. Then go back to step (b) with now $x_2 \in \mathbf{Y}_2$ as parameter and $(x_3, \ldots, x_n)$ as variables, to obtain a point $\tilde{x}_2 \in \mathbf{Y}_2$, etc. until a point $\tilde{\mathbf{x}} \in \prod_{k=1}^n \mathbf{Y}_k$ is obtained. Notice that now $\tilde{\mathbf{x}} \in \mathbf{K}$ because $\mathbf{K}$ is convex. Then proceed as before with $\tilde{\mathbf{x}}$ being the initial guess of a local minimization algorithm to obtain a local minimizer $\hat{\mathbf{x}} \in \mathbf{K}$ of $\mathbf{P}$.

The computational complexity before the local optimization procedure is less than solving n times the i-th semidefinite relaxation in the (J+M)-hierarchy (which is itself of same order as the i-th semidefinite relaxation in the hierarchy of semidefinite relaxations (2.16) defined in Chapter 2, i.e., a polynomial in the input size of $\mathbf{P}$ (with an a priori fixed precision).

Importantly the latter algorithm can be applied to 0/1 programs for which existence of a feasible solution is easy to decide. For instance, MAXCUT, k-CLUSTER and 0/1 KNAPSACK are typical examples of such problems.

An outline of material in this chapter is as follows. In Section 2, we briefly discuss the parametric polynomial optimization problem and a related infinite-dimensional program. In Section 3 and 4, we show how to use results from parametric optimization to define the "joint+marginal" approach and the "joint+marginal" algorithm for helping solve problem $\mathbf{P}$. Finally in the last section, we present numerical experiments for some non convex NP-hard optimization problems (both continuous and 0/1 problems).

4.2 Parametric optimization

Most of the material of this section is taken from [45]. We recall here some notation: Let $\mathbb{R}[\mathbf{x}, \mathbf{y}]$ denote the ring of polynomials in the variables $\mathbf{x} = (x_1, \ldots, x_n)$, and the variables $\mathbf{y} = (y_1, \ldots, y_p)$, whereas $\mathbb{R}[\mathbf{x}, \mathbf{y}]_d$ denotes its subspace of polynomials of degree at most d. Let $\Sigma[\mathbf{x}, \mathbf{y}] \subset \mathbb{R}[\mathbf{x}, \mathbf{y}]$ denote the subset of polynomials that are sums of squares (in short SOS).

4.2.1 Parametric polynomial optimization

Let $\mathbf{Y} \subset \mathbb{R}^p$ be a compact set, called the *parameter* set, and let $f, h_j \in \mathbb{R}[\mathbf{x}, \mathbf{y}]$, $j = 1, \ldots, m$. Let $\mathbf{K} \subset \mathbb{R}^n \times \mathbb{R}^p$ be the basic closed semi-algebraic set:

$$\mathbf{K} := \{(\mathbf{x}, \mathbf{y}) \,:\, \mathbf{y} \in \mathbf{Y}\,;\; h_j(\mathbf{x}, \mathbf{y}) \,\geq\, 0,\; j = 1, \ldots, m\} \tag{4.2}$$

and for each $\mathbf{y} \in \mathbf{Y}$, let

$$\mathbf{K}_{\mathbf{y}} \;:=\; \{\, \mathbf{x} \in \mathbb{R}^n \;:\; (\mathbf{x}, \mathbf{y}) \,\in\, \mathbf{K}\,\}. \tag{4.3}$$

For each $\mathbf{y} \in \mathbf{Y}$, fixed, consider the optimization problem:

$$(\mathbf{P}_{\mathbf{y}}) \quad J(\mathbf{y}) \;:=\; \inf_{\mathbf{x}} \{\, f(\mathbf{x}, \mathbf{y}) \;:\; (\mathbf{x}, \mathbf{y}) \,\in\, \mathbf{K}\,\}. \tag{4.4}$$

The interpretation is as follows: $\mathbf{Y}$ is a set of parameters and for each instance $\mathbf{y} \in \mathbf{Y}$ of the parameter, one wishes to compute an optimal *decision* vector $\mathbf{x}^*(\mathbf{y})$ that solves problem (4.4). Let φ be a Borel probability measure on $\mathbf{Y}$, with a positive density with respect to the Lebesgue measure on $\mathbb{R}^p$ (or with respect to the counting measure if $\mathbf{Y}$ is discrete). For instance

$$\varphi(B) \;:=\; \left(\int_{\mathbf{Y}} d\mathbf{y}\right)^{-1} \int_{\mathbf{Y}\cap B} d\mathbf{y}, \qquad \forall B \in \mathcal{B}(\mathbb{R}^p),$$

is uniformly distributed on $\mathbf{Y}$. Sometimes, e.g. in the context of optimization with data uncertainty, φ is already specified. The idea is to use φ (or more precisely, its moments) to get information on the distribution of optimal solutions $\mathbf{x}^*(\mathbf{y})$ of $\mathbf{P}_{\mathbf{y}}$, viewed as random vectors. In this section we assume that for every $\mathbf{y} \in \mathbf{Y}$, the set $\mathbf{K}_{\mathbf{y}}$ in (4.3) is nonempty.

4.2.2 A related infinite-dimensional linear program

Let $\mathbf{M}(\mathbf{K})$ be the set of finite Borel probability measures on $\mathbf{K}$, and consider the following infinite-dimensional linear program $\mathbf{P}_0$:

$$\rho \;:=\; \inf_{\mu \in \mathbf{M}(\mathbf{K})} \left\{ \int_{\mathbf{K}} f\, d\mu \;:\; \pi\mu \;=\; \varphi \right\}, \tag{4.5}$$

where $\pi\mu$ denotes the marginal of μ on $\mathbb{R}^p$, that is, $\pi\mu$ is a probability measure on $\mathbb{R}^p$ defined by $\pi\mu(B) := \mu(\mathbb{R}^n \times B)$ for all $B \in \mathcal{B}(\mathbb{R}^p)$. The dual of $\mathbf{P}_0$ is the following infinite-dimensional linear program:

$$\begin{array}{rcl} \rho^* := \sup\limits_{p\in\mathbb{R}[\mathbf{y}]} & \displaystyle\int_{\mathbf{Y}} p(\mathbf{y})\, d\varphi(\mathbf{y}) & \\ & f(\mathbf{x},\mathbf{y}) - p(\mathbf{y}) \geq 0 \quad \forall (\mathbf{x},\mathbf{y}) \in \mathbf{K}. \end{array} \tag{4.6}$$

Theorem 4.2.1 ([45]). *Let both $\mathbf{Y} \subset \mathbb{R}^p$ and $\mathbf{K}$ in (4.2) be compact and assume that for every $\mathbf{y} \in \mathbf{Y}$, the set $\mathbf{K}_{\mathbf{y}} \subset \mathbb{R}^n$ in (4.3) is nonempty. Let $\mathbf{P}_0$ be the optimization problem (4.5) and let $\mathbf{X}^*_{\mathbf{y}} := \{\mathbf{x} \in \mathbb{R}^n \,:\, f(\mathbf{x},\mathbf{y}) = J(\mathbf{y})\}$, $\mathbf{y} \in \mathbf{Y}$. Then:*

(a) $\rho = \displaystyle\int_{\mathbf{Y}} J(\mathbf{y})\, d\varphi(\mathbf{y})$ *and $\mathbf{P}_0$ has an optimal solution μ^*.*

(b) *Assume that for φ-almost $\mathbf{y} \in \mathbf{Y}$, the set of minimizers of $\mathbf{X}^*_{\mathbf{y}}$ is the singleton $\{\mathbf{x}^*(\mathbf{y})\}$ for some $\mathbf{x}^*(\mathbf{y}) \in \mathbf{K}_{\mathbf{y}}$. Then there is a measurable mapping $g : \mathbf{Y} \to \mathbf{K}_{\mathbf{y}}$ such that*

$$\begin{array}{rcl} g(\mathbf{y}) & = & \mathbf{x}^*(\mathbf{y}) \;\; \textit{for every} \;\; \mathbf{y} \in \mathbf{Y} \\ \rho & = & \displaystyle\int_{\mathbf{Y}} f(g(\mathbf{y}),\mathbf{y})\, d\varphi(\mathbf{y}), \end{array} \tag{4.7}$$

and for every $\alpha \in \mathbb{N}^n$, and $\beta \in \mathbb{N}^p$:

$$\int_{\mathbf{K}} \mathbf{x}^\alpha \mathbf{y}^\beta \, d\mu^*(\mathbf{x},\mathbf{y}) \;=\; \int_{\mathbf{Y}} \mathbf{y}^\beta \, g(\mathbf{y})^\alpha \, d\varphi(\mathbf{y}). \tag{4.8}$$

(c) *There is no duality gap between (4.5) and (4.6), i.e. $\rho = \rho^*$, and if $(p_i)_{i\in\mathbb{N}} \subset \mathbb{R}[\mathbf{y}]$ is a maximizing sequence of (4.6) then:*

$$\int_{\mathbf{Y}} \mid J(\mathbf{y}) - p_i(\mathbf{y}) \mid d\varphi(\mathbf{y}) \;\to\; 0 \quad \textit{as } i \to \infty. \tag{4.9}$$

Moreover, define the functions $(\tilde{p}_i)$ as follows: $\tilde{p}_0 := p_0$, and

$$\mathbf{y} \mapsto \tilde{p}_i(\mathbf{y}) \;:=\; \max\,[\,\tilde{p}_{i-1}(\mathbf{y}), p_i(\mathbf{y})\,], \quad i = 1, 2, \ldots$$

Then $\tilde{p}_i \to J(\cdot)$, φ-almost uniformly[1].

An optimal solution μ^* of $\mathbf{P}_0$ encodes *all* information on the optimal solutions $\mathbf{x}^*(\mathbf{y})$ of $\mathbf{P}_{\mathbf{y}}$. Moreover from Theorem 4.2.1(c), any optimal or nearly optimal solution of $\mathbf{P}^*_0$ provides us with some polynomial lower approximation of the optimal value function $\mathbf{y} \mapsto J(\mathbf{y})$ that converges to $J(\cdot)$ in the $L_1(\varphi)$ norm, and one may also obtain a piecewise polynomial approximation that converges to $J(\cdot)$, φ-almost uniformly. As

[1] A sequence (g_n) on a measure space $(\mathbf{Y}, \mathcal{B}(\mathbf{Y}), \varphi)$ converges to g, *φ-almost uniformly*, if and only if for every $\epsilon > 0$, there is a set $A \in \mathcal{B}(\mathbf{Y})$ such that $\varphi(A) < \epsilon$ and $g_n \to g$, uniformly on $\mathbf{Y} \setminus A$.

a consequence of the Theorem 4.2.1, the dual of each semidefinite relaxation $\mathbf{Q}_i$ provides a polynomial $q_i \in \mathbb{R}[\mathbf{y}]$ bounded above by $J(\mathbf{y})$, and as $i \to \infty$, the sequence $(\tilde{q}_i)$ with $\mathbf{y} \mapsto \tilde{q}_i(\mathbf{y}) := \max_{\ell=1,\dots i} q_\ell(\mathbf{y})$, converges φ-almost uniformly to the optimal value function J. This last property is the rationale behind the heuristic developed below.

4.3 A “joint+marginal" approach

With $\{f, (g_j)_{j=1}^m\} \subset \mathbb{R}[\mathbf{x}]$, let $\mathbf{K} \subset \mathbb{R}^n$ be the basic compact semi-algebraic set

$$\mathbf{K} := \{\mathbf{x} \in \mathbb{R}^n \;:\; g_j(\mathbf{x}) \geq 0,\; j = 1, \dots, m\}, \tag{4.10}$$

and consider the polynomial optimization problem (2.1) with optimal value f^*.

Let $\mathbf{Y}_k \subset \mathbb{R}$ be some interval $[\underline{x}_k, \overline{x}_k]$, assumed to be contained in the orthogonal projection of $\mathbf{K}$ into the x_k-coordinate axis.

For instance when the g_j's are affine (so that $\mathbf{K}$ is a convex polytope), $\underline{x}_k$ (resp. $\overline{x}_k$) solves the linear program min (resp max) $\{x_k : \mathbf{x} \in \mathbf{K}\}$. Similarly, when $\mathbf{K}$ is convex and defined by concave polynomials, one may obtain $\underline{x}_k$ and $\overline{x}_k$, up to (arbitrary) fixed precision. In many cases, (upper and lower) bound constraints on the variables are already part of the problem definition.

Let φ_k be the probability measure uniformly distributed on $\mathbf{Y}_k$; hence its moments (β_ℓ), $\ell \in \mathbb{N}$, are given by:

$$\beta_\ell = \int_{\underline{x}_k}^{\overline{x}_k} x^\ell d\varphi_k(x) \;=\; \frac{\overline{x}_k^{\ell+1} - \underline{x}_k^{\ell+1}}{(\ell+1)(\overline{x}_k - \underline{x}_k)} \tag{4.11}$$

for every $\ell = 0, 1, \dots$. Define the following parametric polynomial program in $n-1$ variables:

$$J^k(y) \;=\; \min_{\mathbf{x}} \{f(\mathbf{x}) \;:\; \mathbf{x} \in \mathbf{K};\;\; x_k = y\}, \tag{4.12}$$

or, equivalently $J^k(y) = \min \{f(\mathbf{x}) \;:\; \mathbf{x} \in \mathbf{K}_y\}$, where for every $y \in \mathbf{Y}$:

$$\mathbf{K}_y := \{\mathbf{x} \in\; \mathbf{K};\, x_k = y\}. \tag{4.13}$$

Observe that by definition, $f^* = \min_x \{J^k(x) : x \in \mathbf{Y}_k\}$, and $\mathbf{K}_y \neq \emptyset$ whenever $y \in \mathbf{Y}_k$, where $\mathbf{Y}_k$ is the orthogonal projection of $\mathbf{K}$ into the x_k-coordinate axis.

Semidefinite relaxations

To compute (or at least approximate) the optimal value ρ of problem $\mathbf{P}_0$ in (4.5) associated with the parametric optimization problem (4.12), we now provide a hierarchy of semidefinite relaxations in the spirit of those defined in Chapter 2. Let $v_j := \lceil(\deg \mathrm{g_j})/2\rceil$, $j = 1, \ldots, m$, and for $i \geq \max_j v_j$, consider the semidefinite program:

$$\begin{aligned} \rho_{ik} = \quad & \inf_{\mathbf{z}} \quad L_{\mathbf{z}}(f) \\ & \text{s.t.} \quad \mathbf{M}_i(\mathbf{z}) \succeq 0,\ \mathbf{M}_{i-v_j}(g_j\,\mathbf{z}) \succeq 0, \quad j = 1, \ldots, m \\ & \qquad\ \ L_{\mathbf{z}}(x_k^\ell) = \beta_\ell, \quad \ell = 0, 1, \ldots 2i, \end{aligned} \tag{4.14}$$

where (β_ℓ) is defined in (4.11), and $\mathbf{z} = (z_\alpha)$, $\alpha \in \mathbb{N}^n$.

We call (4.14) the *parametric semidefinite relaxation* of $\mathbf{P}$ with parameter $y = x_k$. Observe that without the “moment" constraints $L_{\mathbf{z}}(x_k^\ell) = \beta_\ell$, $\ell = 1, \ldots 2i$, the semidefinite program (4.14) is a relaxation of $\mathbf{P}$ and if $\mathbf{K}$ is compact and Assumption 1 holds, its corresponding optimal value f_i^* converges to f^* as $k \to \infty$; see Chapter 2. Letting $g_0 \equiv 1$, the dual of (4.14) reads:

$$\begin{aligned} \rho_{ik}^* = \quad & \sup_{\lambda,(\sigma_j)} \sum_{\ell=0}^{2i} \lambda_\ell\,\beta_\ell \\ \text{s.t.} \quad & f(\mathbf{x}) - \sum_{\ell=0}^{2i} \lambda_\ell x_k^\ell = \sum_{j=0}^{m} \sigma_j\, g_j \\ & \sigma_j \in \Sigma[\mathbf{x}], \quad 0 \leq j \leq m; \\ & \deg \sigma_j g_j \leq 2i, \quad 0 \leq j \leq m. \end{aligned} \tag{4.15}$$

Equivalently, recall that $\mathbb{R}[x_k]_{2i}$ is the space of univariate polynomials of degree at most $2i$, and observe that in (4.15), the objective reads

$$\sum_{\ell=0}^{2i} \lambda_\ell\,\beta_\ell = \int_{\mathbf{Y}_k} p_i(y) d\varphi_k(y),$$

where $p_i \in \mathbb{R}[x_k]_{2i}$ is the univariate polynomial $x_k \mapsto p_i(x_k) := \sum_{\ell=0}^{2i} \lambda_\ell x_k^\ell$. Then equivalently, the above dual may be rewritten as:

$$\begin{aligned}\rho^*_{ik} = & \sup_{p_i,(\sigma_j)} \int_{\mathbf{Y}_k} p_i d\varphi_k \\ \text{s.t.} \quad & f - p_i = \sum_{j=0}^{m} \sigma_j \, g_j \\ & p_i \in \mathbb{R}[x_k]_{2i};\ \sigma_j \in \Sigma[\mathbf{x}], \quad 0 \leq j \leq m; \\ & \deg \sigma_j g_j \leq 2i, \quad 0 \leq j \leq m. \end{aligned} \tag{4.16}$$

Theorem 4.3.1. *Let* $\mathbf{K}$ *be as in (4.10) and let Assumption 1 hold. Let the interval* $\mathbf{Y}_k \subset \mathbb{R}$ *be the orthogonal projection of* $\mathbf{K}$ *into the* x_k*-coordinate axis, and let* φ_k *be the probability measure, uniformly distributed on* $\mathbf{Y}_k$*. Assume that* $\mathbf{K}_y$ *(for all* y*) in (4.13) is not empty, let* $y \mapsto J^k(y)$ *be as in (4.12) and consider the semidefinite relaxations (4.14)-(4.16). Then as* $i \to \infty$*:*

(a) $\rho_{ik} \uparrow \int_{\mathbf{Y}_k} J^k d\varphi_k$ *and* $\rho^*_{ik} \uparrow \int_{\mathbf{Y}_k} J^k d\varphi_k$

(b) *Let* $(p_i, (\sigma^i_j))$ *be a nearly optimal solution of (4.16), e.g. such that* $\int_{\mathbf{Y}_k} p_i d\varphi_k \geq \rho^*_{ik} - 1/i$*. Then* $p_i(y) \leq J^k(y)$ *for all* $y \in \mathbf{Y}_k$*, and*

$$\int_{\mathbf{Y}_k} |J^k(y) - p_i(y)| \, d\varphi_k(y) \ \to \ 0, \quad \text{as } i \to \infty. \tag{4.17}$$

Moreover, if one defines $\tilde{p}_0 := p_0$*, and*

$$\mathbf{y} \mapsto \tilde{p}_i(y) \ := \ \max\left[\, \tilde{p}_{i-1}(y), p_i(y) \,\right], \quad i = 1, 2, \ldots,$$

then $\tilde{p}_i(y) \uparrow J^k(y)$*, for* φ_k*-almost all* $y \in \mathbf{Y}_k$*, and so* $\tilde{p}_i \to J^k$*,* φ_k*-almost uniformly on* $\mathbf{Y}_k$*.*

For a detailed proof the interested reader is referred to e.g. [45].

4.4 A "joint+marginal" algorithm

4.4.1 A "joint+marginal" algorithm for the general nonconvex case

Theorem 4.3.1 provides a rationale for the following (J+M)-algorithm in the general nonconvex case. In what follows we use the primal and dual semidefinite relaxations (4.14)-(4.15) with index i *fixed*.

ALGO 1: (J+M)-algorithm: non convex $\mathbf{K}$, relaxation i

Set $k = 1$**;**
Step k: **Input:** $\mathbf{K}$, f, and the orthogonal projection $\mathbf{Y}_k = [\underline{x}_k, \overline{x}_k]$ of $\mathbf{K}$ into the x_k-coordinate axis, with associated probability measure φ_k, uniformly distributed on $\mathbf{Y}_k$.
Ouput: $\tilde{x}_k \in \mathbf{Y}_k$.
Solve the semidefinite program (4.16) and from an optimal (or nearly optimal) solution $(p_i, (\sigma_j))$ of (4.16), get a global minimizer $\tilde{x}_k$ of the univariate polynomial p_i on $\mathbf{Y}_k$.
If $k = n$ stop and output $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_n)$, otherwise set $k = k + 1$ and repeat.

Of course, in general the vector $\tilde{\mathbf{x}} \in \mathbb{R}^n$ does not belong to $\mathbf{K}$. Therefore a final step consists of computing a local minimum $\hat{\mathbf{x}} \in \mathbf{K}$, by using some local minimization algorithm starting with the (unfeasible) initial point $\tilde{\mathbf{x}}$. But again, the rationale behind this algorithm is Theorem 4.3.1 which suggests that if the order i of the relaxation (in the algorithm i is fixed) is sufficiently large then the cost associated with $\tilde{\mathbf{x}}$ should be close to the optimal cost.

Also note that when $\mathbf{K}$ is not convex, the determination of bounds $\underline{x}_k$ and $\overline{x}_k$ for the interval $\mathbf{Y}_k$ may not be easy, and so one might be forced to use a subinterval $\mathbf{Y}'_k \subseteq \mathbf{Y}_k$ with conservative (but computable) bounds $\underline{x}'_k \geq \underline{x}_k$ and $\overline{x}'_k \leq \overline{x}_k$.

Remark 4.4.1. *Theorem 4.3.1 assumes that for every $y \in \mathbf{Y}_k$, the set $\mathbf{K}_y$ in (4.13) is not empty, which is the case if $\mathbf{K}$ is connected. If $\mathbf{K}_y = \emptyset$ for y in some open subset of $\mathbf{Y}_k$, then the semidefinite relaxation (4.14) has no solution ($\rho_{ik} = +\infty$), in which case one proceeds by dichotomy on the interval $\mathbf{Y}_k$ until $\rho_{ik} < \infty$. That is, with $z := (\underline{x}_k + \overline{x}_k)/2$, consider the subintervals $\mathbf{Y}^1_k := [\underline{x}_k, z]$ and $\mathbf{Y}^2_k := [z, \overline{x}_k]$. Solve (4.14) where the (β_ℓ) in (4.11) are updated according to $\mathbf{Y}^1_k$ (resp. $\mathbf{Y}^2_k$) to obtain ρ^1_{ik} (resp. ρ^2_{ik}). If $\rho^s_{ik} < \infty$ for some s, stop and proceed with $\mathbf{Y}^s_k$ instead of $\mathbf{Y}_k$, otherwise choose randomly $\mathbf{Y}^1_k$ or $\mathbf{Y}^2_k$ and iterate.*

4.4.2 A "joint+marginal" algorithm when K is convex

In this section, we now assume that the feasible set $\mathbf{K} \subset \mathbb{R}^n$ of problem $\mathbf{P}$ is convex (and compact). The idea is to compute $\tilde{x}_1$ as in **ALGO 1** and then repeat the procedure but now for the $(n-1)$-variable problem $\mathbf{P}(\tilde{x}_1)$ which is problem $\mathbf{P}$ in which the variable x_1 is *fixed* at the value $\tilde{x}_1$. This alternative is guaranteed to work if $\mathbf{K}$ is convex (but not always if $\mathbf{K}$ is not convex).

For every $j \geq 2$, denote by $\mathbf{x}_j \in \mathbb{R}^{n-j+1}$ the vector $(x_j, \ldots, x_n)$, and by $\tilde{\mathbf{x}}_{j-1} \in \mathbb{R}^{j-1}$

the vector $(\tilde{x}_1, \ldots, \tilde{x}_{j-1})$ (and so $\tilde{\mathbf{x}}_1 = \tilde{x}_1$).

Let the interval $\mathbf{Y}_1 \subset \mathbb{R}$ be the orthogonal projection of $\mathbf{K}$ into the x_1-coordinate axis. For every $\tilde{x}_1 \in \mathbf{Y}_1$, let the interval $\mathbf{Y}_2(\tilde{\mathbf{x}}_1) \subset \mathbb{R}$ be the orthogonal projection of the set $\mathbf{K} \cap \{\mathbf{x} : x_1 = \tilde{x}_1\}$ into the x_2-coordinate axis. Similarly, given $\tilde{\mathbf{x}}_2 \in \mathbf{Y}_1 \times \mathbf{Y}_2(\tilde{\mathbf{x}}_1)$, let the interval $\mathbf{Y}_3(\tilde{\mathbf{x}}_2) \subset \mathbb{R}$ be the orthogonal projection of the set $\mathbf{K} \cap \{\mathbf{x} : x_1 = \tilde{x}_1;\, x_2 = \tilde{x}_2\}$ into the x_3-coordinate axis, and etc. in the obvious way.

For every $k = 2, \ldots, n$, and $\tilde{\mathbf{x}}_{k-1} \in \mathbf{Y}_1 \times \mathbf{Y}_2(\tilde{\mathbf{x}}_1) \cdots \times \mathbf{Y}_{k-1}(\tilde{\mathbf{x}}_{k-2})$, let $\tilde{f}_k(\mathbf{x}_k) := f((\tilde{\mathbf{x}}_{k-1}, \mathbf{x}_k))$, and $\tilde{g}_j^k(\mathbf{x}_k) := g_j((\tilde{\mathbf{x}}_{k-1}, \mathbf{x}_k))$, $j = 1, \ldots, m$. Similarly, let

$$\begin{aligned} \mathbf{K}_k(\tilde{\mathbf{x}}_{k-1}) &:= \{\mathbf{x}_k \,:\, \tilde{g}_j^k(\mathbf{x}_k) \geq 0,\ j = 1, \ldots, m\}, \\ &= \{\mathbf{x}_k \,:\, (\tilde{\mathbf{x}}_{k-1}, \mathbf{x}_k) \in \mathbf{K}\}, \end{aligned} \tag{4.18}$$

and consider the problem:

$$\mathbf{P}(\tilde{\mathbf{x}}_{k-1}) : \quad \min\{\tilde{f}_k(\mathbf{x}_x) \,:\, \mathbf{x}_x \in \mathbf{K}_j(\tilde{\mathbf{x}}_{k-1})\}, \tag{4.19}$$

i.e. the original problem $\mathbf{P}$ where the variable x_ℓ is fixed at the value $\tilde{x}_\ell$, for every $\ell = 1, \ldots, k-1$.

Write $\mathbf{Y}_j(\tilde{\mathbf{x}}_{k-1}) = [\underline{x}_k, \overline{x}_k]$, and let φ_k be the probability measure uniformly distributed on $\mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})$.

Let $\mathbf{z}$ be a sequence indexed in the monomial basis of $\mathbb{R}[\mathbf{x}_k]$. With index i, fixed, the parametric semidefinite relaxation (4.14) with parameter x_k, associated with problem $\mathbf{P}(\tilde{\mathbf{x}}_{k-1})$, reads:

$$\begin{aligned} \rho_{ik} = \inf_{\mathbf{z}} \quad & L_{\mathbf{z}}(\tilde{f}_k) \\ \text{s.t.} \quad & \mathbf{M}_i(\mathbf{z}),\ \mathbf{M}_{i-v_j}(\tilde{g}_j^k\,\mathbf{z}) \succeq 0, \quad j = 1, \ldots, m \\ & L_{\mathbf{z}}(x_k^\ell) = \beta_\ell, \quad \ell = 0, 1, \ldots, 2i, \end{aligned} \tag{4.20}$$

where (β_ℓ) is defined in (4.11). Its dual is the semidefinite program (with $\tilde{g}_0^k \equiv 1$)):

$$\begin{aligned} \rho_{ik}^* \quad = \quad & \sup_{p_i, (\sigma_j)} \int_{\mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})} p_i d\varphi_k \qquad (4.21) \\ \text{s.t.} \quad & \tilde{f}_k - p_i = \sigma_0 + \sum_{j=1}^m \sigma_j\, \tilde{g}_j^k \\ & p_i \in \mathbb{R}[x_k]_{2i},\ \sigma_j \in \Sigma[\mathbf{x}_k], \quad j = 0, \ldots, m \\ & \deg \sigma_j \tilde{g}_j^k \leq 2i, \quad j = 0, \ldots, m. \end{aligned}$$

The important difference between (4.14) and (4.20) is the *size* of the corresponding semidefinite programs, since $\mathbf{z}$ in (4.14) (resp. in (4.20)) is indexed in the canonical basis of $\mathbb{R}[\mathbf{x}]$ (resp. $\mathbb{R}[\mathbf{x}_k]$).

The (J+M)-algorithm for K convex

Recall that the order i of the semidefinite relaxation is fixed. The (J+M)-algorithm consists of n steps. At step k of the algorithm, the vector $\tilde{\mathbf{x}}_{k-1} = (\tilde{x}_1, \ldots, \tilde{x}_{k-1})$ (already computed) is such that $\tilde{x}_1 \in \mathbf{Y}_1$ and $\tilde{x}_\ell \in \mathbf{Y}_\ell(\tilde{\mathbf{x}}_{\ell-1})$ for every $\ell = 2, \ldots, k-1$, and so the set $\mathbf{K}_k(\tilde{\mathbf{x}}_{k-1})$ is a nonempty compact convex set.

ALGO 2: (J+M)-algorithm: convex K, relaxation i

Set $k = 1$**;**
Step $k \geq 1$: **Input:** For $k = 1$, $\tilde{\mathbf{x}}_0 = \emptyset$, $\mathbf{Y}_1(\tilde{\mathbf{x}}_0) = \mathbf{Y}_1$; $\mathbf{P}(\tilde{\mathbf{x}}_0) = \mathbf{P}$, $f_1 = f$ and $\tilde{g}^1_j = g_j$, $j = 1, \ldots, m$.
For $k \geq 2$, $\tilde{\mathbf{x}}_{k-1} \in \mathbf{Y}_1 \times \mathbf{Y}_2(\tilde{x}_1) \cdots \times \mathbf{Y}_{k-1}(\tilde{x}_{k-2})$.
Output: $\tilde{\mathbf{x}}_k = (\tilde{\mathbf{x}}_{k-1}, \tilde{x}_k)$ with $\tilde{x}_k \in \mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})$.
Consider the parametric semidefinite relaxations (4.20)-(4.21) with parameter x_k, associated with problem $\mathbf{P}(\tilde{\mathbf{x}}_{k-1})$ in (4.19).

- From an optimal solution of (4.21), extract the univariate polynomial $x_k \mapsto p_i(x_k) := \sum_{\ell=0}^{2i} \lambda^*_\ell x_k^\ell$.

- Get a global minimizer $\tilde{x}_k$ of p_i on the interval $\mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1}) = [\underline{x}_k, \overline{x}_k]$, and set $\tilde{\mathbf{x}}_k := (\tilde{\mathbf{x}}_{k-1}, \tilde{x}_k)$.

If $k = n$ stop and ouput $\tilde{\mathbf{x}} \in \mathbf{K}$, otherwise set $k = k + 1$ and repeat.

As $\mathbf{K}$ is convex, $\tilde{\mathbf{x}} \in \mathbf{K}$ and one may stop. A refinement is to now use $\tilde{\mathbf{x}}$ as the initial guess of a local minimization algorithm to obtain a local minimizer $\hat{\mathbf{x}} \in \mathbf{K}$ of $\mathbf{P}$. In view of Theorem 4.3.1, the larger the index i of the relaxations (4.20)-(4.21), the better the values $f(\tilde{\mathbf{x}})$ and $f(\hat{\mathbf{x}})$.

4.4.3 The (J+M)-algorithm for some 0/1 programs

Of course, **ALGO 2** can also be used when $\mathbf{K}$ is not convex. However, it may happen that at some stage k, the semidefinite relaxation (4.20) may be infeasible because $J^k(y)$ is infinite for some values of $y \in \mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})$. This is because the feasible set $\mathbf{K}(\tilde{\mathbf{x}}_{k-1})$ in (4.18) may be disconnected.

However there is an interesting class of problems with non convex set $\mathbf{K}$ for which **ALGO 2** can be applied. It consist of 0/1 programs for which feasibility is easy to test. Among them let us cite the celebrated MAXCUT, k-CLUSTER and 0/1 KNAPSACK problems.

In this case the set $\mathbf{K}$ is contained in the hyper cube $\{0,1\}^n$ (or $\{-1,1\}^n$ for MAXCUT), i.e., the set $\mathbf{K}$ is now defined by:

$$\mathbf{K} := \{\mathbf{x}\in\mathbb{R}^n \,:\, g_j(\mathbf{x})\geq 0,\ j=1,\ldots,m;\quad u_k(\mathbf{x})=0,\ k=1,\ldots,n\},$$

where $\mathbf{x}\mapsto u_k(\mathbf{x}) = x_k^2 - x_k$, $k=1,\ldots,n$. The parameter set $\mathbf{Y}=\{0,1\}^n$ and $\mathbf{Y}_k=\{0,1\}$ for all $k=1,\ldots,n$. As a probability measure φ on $\mathbf{Y}_k$ we take p and $1-p$ for some fixed $p\in(0,1)$ (e.g. $p=1/2$).

The semidefinite relaxation (4.20) now reads

$$\begin{aligned}\rho_{ik} = \ \inf_{\mathbf{z}}\ \ & L_{\mathbf{z}}(\tilde{f}_k)\\ \text{s.t.}\ \ & \mathbf{M}_i(\mathbf{z})\succeq 0,\ \mathbf{M}_{i-v_j}(\tilde{g}_j^k\,\mathbf{z})\succeq 0,\quad j=1,\ldots,m\\ & \mathbf{M}_{i-1}(u_\ell\,\mathbf{z})=0,\ \ell=k,\ldots,n;\quad L_{\mathbf{z}}(1)=1,\ L_{\mathbf{z}}(x_k)=p,\end{aligned}\tag{4.22}$$

Indeed the additional moment constraints $L_{\mathbf{z}}(x_k^\ell)=\beta_\ell$ of (4.20) are redundant since $L_{\mathbf{z}}(x_k^{2\ell}) = L_{\mathbf{z}}(1) = 1$ and $L_{\mathbf{z}}(x_k^{2\ell+1}) = L_{\mathbf{z}}(x_k) = p$. This semidefinite program (4.22) can be simplified to work modulo the ideal $\langle x_k^2-x_k,\cdots,x_n^2-x_n\rangle$. Get rid of the constraints $\mathbf{M}_{i-1}(u_\ell\,\mathbf{z})=0$, $\ell=k,\ldots,n$, and replace every moment variable z_α with z_β where $\beta_\ell=1$ if $\alpha_\ell>0$, and $\beta_\ell=0$ if $\alpha_\ell=0$. Next, the row and column of $\mathbf{M}_i(\mathbf{z})$ associated with the monomial $\mathbf{x}^\alpha$ is identical to that associated with $\mathbf{x}^\beta$ and can be deleted; same thing with the localizing matrices $\mathbf{M}_{i-v_j}(g_j\mathbf{z})$. The dual of (4.22) reads:

$$\begin{aligned}\rho_{ik}^* = \sup_{\lambda,(\sigma_\ell)}\ \ & \lambda_0+p\lambda_1\\ \text{s.t.}\ \ & \tilde{f}_k-(\lambda_0+\lambda_1x_k) = \sigma_0+\sum_{j=1}^m\sigma_j\,\tilde{g}_j^k+\sum_{\ell=k}^n\psi_\ell\,u_\ell\\ & \sigma_j\in\Sigma[\mathbf{x}_k],\ \psi_\ell\in\mathbb{R}[\mathbf{x}_k],\quad j=0,\ldots,m;\ \ell=k,\ldots,n\\ & \deg\sigma_j\,\tilde{g}_j^k\leq 2i,\ \deg\psi_\ell\leq 2i-2,\quad j=0,\ldots,m;\ \ell=k,\ldots,n.\end{aligned}\tag{4.23}$$

So the polynomial p_i of (4.15) is now an affine polynomial $x_k\mapsto p_i(x_k) := \lambda_0+\lambda_1x_k$ and the global minimum of p_i on $\mathbf{Y}_k=\{0,1\}$ is just $x_k=0$ if $\lambda_1\geq 0$ and $x_k=1$

otherwise.

Since we are not in the convex setting anymore, it remains to check whether after having chosen $x_k = 0$ (or $x_k = 1$) the remaining problem $\mathbf{P}(\tilde{\mathbf{x}}_k)$ with variables $(x_{k+1}, \ldots, x_n)$ has at least a solution. And this why this approach is possible for $0/1$ problems for which feasibility is easy to detect. For several problems like e.g., MAX-CUT, KNAPSACK or k-CLUSTER, whenever $\mathbf{P}(\tilde{\mathbf{x}}_k)$ is feasible, it has an obvious feasible solution denoted $\mathbf{s}_{k+1} \in \{0,1\}^{n-k}$.

For instance, the $0/1$ knapsack problem $\mathbf{P}(\tilde{\mathbf{x}}_k)$: $\min_{\mathbf{x}_j} \{\mathbf{c}_j'\mathbf{x}_j : \sum_{\ell=k+1}^n a_\ell \, x_\ell \leq b(\tilde{\mathbf{x}}_k)\}$, has the obvious feasible solution $\mathbf{s}_{k+1} = 0$ whenever $b(\tilde{\mathbf{x}}_k) \geq 0$, and no solution otherwise. Similarly, the k-cluster problem $\mathbf{P}(\tilde{\mathbf{x}}_k)$: $\min_{\mathbf{x}_j} \{\mathbf{x}_j' Q \mathbf{x}_j : \sum_{\ell=k+1}^n x_\ell = k(\tilde{\mathbf{x}}_k)\}$ (with $k(\tilde{\mathbf{x}}_k) \in \mathbb{Z}$), has no solution if $k(\tilde{\mathbf{x}}_k) < 0$ or $k(\tilde{\mathbf{x}}_k) > n - k$, has the obvious optimal solution $\mathbf{s}_{k+1} = 0$ if $k(\tilde{\mathbf{x}}_k) = 0$, and whenever $0 < k(\tilde{\mathbf{x}}_k) \leq n - k - 1$, it has the obvious feasible solution $\mathbf{s}_{k+1} = (s_{(k+1)\ell})$ with $s_{(k+1)\ell} = 1$, $\ell = 1, \ldots, k(\tilde{\mathbf{x}}_k)$, and $s_{(k+1)\ell} = 0$, $\ell = k(\tilde{\mathbf{x}}_k) + 1, \ldots, n$.

So assume that one already knows that $\mathbf{P}(\tilde{\mathbf{x}}_{k-1})$ has at least a feasible solution. One solves (4.22) and determines x_k from a solution of (4.23) as above only when $\mathbf{P}(\tilde{\mathbf{x}}_{k-1})$ has at least a solution $(0, s_{k+1}, \ldots, s_n)$ (if not set $x_k = 1$) and a solution $(1, s_{k+1}', \ldots, s_n')$ (if not then set $x_k = 0$). By construction, after having determined x_k like this, the resulting next problem $\mathbf{P}(\tilde{\mathbf{x}}_k)$ has at least a solution and one may continue the procedure.

Of course an alternative to this procedure of computing x_k from a solution of (4.23) is to proceed like in standard Branch & Bound. That is, compute the value of the i-th SDP relaxation of Chapter 2 associated with the original problem $\mathbf{P}$ and the additional constraint $x_1 = 0$ (resp. $x_1 = 1$), and branch on $x_1 = 0$ or on $x_1 = 1$ depending on the resulting optimal values; then iterate with x_2, etc. But this requires to solve *two* semidefinite relaxations of same size (with only one linear constraint less) instead of one in the parametric approach. Moreover, the parametric approach can easily deal with *groups* of variables (rather than 1 variable) at a time. Indeed, with $s \in \mathbb{N}$ fixed, consider $(x_1, \ldots, x_s) \in \mathbf{Y} := \{0,1\}^s$ with associated probability distribution φ uniformly distributed on $\mathbf{Y}$. Then one now solves the i-th SDP relaxation of the (J+M)-hierarchy applied to problem $\mathbf{P}$ with $n - s$ variables $x_{s+1}, \ldots, x_n$ and parameter $(x_1, \ldots, x_s) \in \{0,1\}^s$. The dual provides a (square free) polynomial map $(x_1, \ldots, x_s) \mapsto J_k(x_1, \ldots, x_s)$ that converges to $J(x_1, \ldots, x_s)$ as k increases. Then one selects a global minimizer of J_k on $\mathbf{Y}$ by inspecting 2^s values, and one iterates the procedure with now a $0/1$ problem with $n - s$ variables, etc.

4.5 Computational results

We have tested the algorithms on a set of difficult global optimization problems taken from Floudas et al. [15]. To solve the semidefinite programs involved in **ALGO 1** and in **ALGO 2**, we have used the GloptiPoly software [29] that implements the hierarchy of semidefinite relaxations defined in Chapter 2.

4.5.1 Computational results for continuous problems

We present computational experiments on some non convex NP-hard optimization problems.

In a first set of examples, the set $\mathbf{K}$ is a convex polytope and the function f is a nonconvex quadratic polynomial $\mathbf{x} \mapsto \mathbf{x}'Q\mathbf{x} + \mathbf{b}'\mathbf{x}$ for some real symmetric matrix Q and vector $\mathbf{b}$. The optimal value obtained using the output $\tilde{\mathbf{x}}$ of **ALGO 2** as initial guess in a local minimization algorithm of the MATLAB toolbox; notice that since $\tilde{\mathbf{x}} \in \mathbf{K}$ one might have stopped with this current feasible solution. As may be seen from Table 4.1, the associated relative error is very small.

In a second set of examples the set $\mathbf{K}$ is not convex and we have used **ALGO 1** to find a minimizer $\tilde{\mathbf{x}}$ (which may not belong to $\mathbf{K}$). Therefore a final step of **ALGO 1** is to use some local minimization algorithm starting with the initial (infeasible) point $\tilde{\mathbf{x}}$. Also in some cases, the determination of bounds $\underline{x}_k$ and $\overline{x}_k$ for the interval $\mathbf{Y}_k$ may not be easy, and so one might be forced to use a subinterval $\mathbf{Y}'_k \subseteq \mathbf{Y}_k$ with conservative (but computable) bounds $\underline{x}'_k \geq \underline{x}_k$ and $\overline{x}'_k \leq \overline{x}_k$ (table 4.2).

Results are displayed in Table 4.1 and Table 4.2 in which the columns respectively stand for the problem's name, the number n of variables, the number m of constraints, the global opitimum f^*, the index i of the semidefinite relaxation, the optimal value obtained by using the output of algorithms as initial guess in a local minimization algorithm of the MATLAB toolbox, and the associated relative error.

For the non convex examples it is worth mentioning that sometimes the output $\tilde{\mathbf{x}}$ (not in $\mathbf{K}$ in general) of **ALGO 1** is far from the final feasible solution $\mathbf{x} \in \mathbf{K}$ and so it is hard to evaluate the importance of the final step of local minimization from the initial infeasible solution $\tilde{\mathbf{x}}$.

Prob	n	m	f^*	i	ALGO 2	rel. error
2.2	5	11	-17	2	-17.00	0%
2.3	6	8	-361.5	1	-361.50	0%
2.6	10	21	-268.01	1	-267.00	0.3%
2.9	10	21	0	1	0.00	0%
2.8C1	20	30	-394.75	1	-385.30	2.4%
2.8C2	20	30	-884.75	1	-871.52	1.5%
2.8C3	20	30	-8695	1	-8681.7	0.15%
2.8C4	20	30	-754.75	1	-754.08	0.09%
2.8C5	20	30	-4150.41	1	-3678.2	11%

Table 4.1: **ALGO 2** for convex set **K**

Prob	n	m	f^*	i	ALGO 1	rel. error
3.2	8	22	7049	1	7049	0%
3.3	5	16	-30665	1	-30665	0%
3.4	6	18	-310	1	-310	0%
5.2.2 (1)	9	24	400	1	400	0%
5.2.2 (2)	9	24	600	1	600	0%
5.2.3 (3)	9	24	750	1	750	0%
5.2.4	9	24	450	1	450	0%
7.2.2	6	17	-0.3746	1	-0.3746	0%
7.2.3	8	22	7049.25	1	7049.25	0%
7.2.5	5	16	10122	1	10122	0%
7.2.6	3	7	-83.254	1	-83.258	1%
8.2.8	6	17	-0.3768	1	-0.3767	0%

Table 4.2: **ALGO 1** for non convex set **K**

4.5.2 Computational results for some 0/1 programs

4.5.2.1 The MAXCUT problem

The celebrated MAXCUT problem formally consists of solving the optimization problem:

$$\mathbf{P}: \quad \max_{\mathbf{x}} \left\{ \frac{1}{2} \sum_{1 \le i < j \le n} Q_{ij} \left(1 - x_i x_j\right) : \mathbf{x} \in \{-1, 1\}^n \right\},$$

for some real symmetric matrix $Q = (Q_{ij}) \in \mathbb{R}^{n \times n}$.

In our sample of randomly generated problems, the entry Q_{ij} of the real symmetric matrix Q is set to zero with probability 1/2 and when different from zero, Q_{ij} is randomly (and independently) generated according to the uniform probability distribution on the interval $[0, 10]$.

We have tested the basic version of the (J+M)-algorithm with $i = 1$ and $p = 1/2$, i.e. solving the SDP-relaxations (4.22)-(4.23), for MAXCUT problems on random graphs with $n = 20, 30, 40$ and 50 variables. For each value of n, we have generated 50 problems (and 20 for $n = 50$). In (4.14) the parameter $p \in (0, 1)$ is set to 0.5. Let $\mathbf{Q}_1$ denote the optimal value of the SDP-relaxation (4.22) (with $i = k = 1$) without the marginal constraint $x_1 = p$, that is, $\mathbf{Q}_1$ is the Shor's relaxation with famous Goemans and Williamson's 0.878 performance guarantee (for maximization and nonnegative weights Q_{ij}). Let $\mathbf{P}_1$ denote the cost of the solution $\mathbf{x} \in \{-1, 1\}^n$ generated by the (J+M)-algorithm[2]. In Table 4.5.2.1 below, we have reported the average relative error $(\mathbf{Q}_1 - \mathbf{P}_1)/|\mathbf{Q}_1|$, which, as one may see, is very small and comparable with the relative error $(\mathbf{Q}_1 - \mathrm{GW})/|\mathbf{Q}_1|$ obtained from the Goemans and Williamson (GW) solution. The latter was obtained by the randomized rounding procedure described in [20] with a sample size of 50 as recommended in [20] for $n \approx 50$ and we used the `schur` MATLAB subroutine to compute eigenvectors. The value of the GW-solution is significantly better than the theoretical bound, in accordance with the computational results displayed in [20]. For $n = 50$ the (J+M)-solution was better than the GW-solution in 7 out of the 20 randomly generated problems. For $n = 40$ (resp. $n = 50$)

[2] $\mathbf{Q}_1$ and $\mathbf{P}_1$ were computed with the GloptiPoly software dedicated to solving the generalized problem of moments [29].

n	20	30	40	50
$(\mathbf{Q}_1 - \mathbf{P}_1)/\lvert\mathbf{Q}_1\rvert$	3.23%	3.28%	3.13%	2.92%
$(\mathbf{Q}_1 - \mathrm{GW})/\lvert\mathbf{Q}_1\rvert$	2.58%	2.60%	2.84%	2.60%

Table 4.3: Relative error for MAXCUT; $p = 0.5$

variables the CPU time was 218s (resp. 945s) whereas it was 232s (resp. 507s) for the GW solution. The reader should keep in mind that the (J+M)-algorithm is not specific to the MAXCUT problem and was run with the smallest possible choice $i = 1$ of the parameter i.

4.5.2.2 The k-CLUSTER problem

We have also tested the (J+M)-algorithm for the k-CLUSTER problem:

$$\mathbf{P}: \qquad \max_{\mathbf{x}} \left\{ \mathbf{x}'Q\mathbf{x} \,:\, \mathbf{x} \in \{0,1\}^n;\, \sum_{\ell=1}^{n} x_\ell \,=\, k \right\}, \tag{4.24}$$

again for some real symmetric matrix $Q = (Q_{ij}) \in \mathbb{R}^{n\times n}$, and some fixed integer $k \in \mathbb{N}$, $1 \leq k < n$. Observe that the constraint $\sum_\ell x_\ell = k$ is linear. Therefore, to take full advantage of the i-th SDP relaxation (4.22) which contains moments z_α of order up to $2i$, one may add the n constraints $x_i(k - \sum_{\ell=1}^n x_\ell) = 0$, $i = 1, \ldots, n$, in the definition (4.24) of $\mathbf{P}$ because they are redundant. However these constraints make the i-th SDP relaxation more constrained. They also correspond to the first-level RLT constraints defined in Sherali and Adams [70].

As for MAXCUT, $\mathbf{Q}_1$ denotes the optimal value of the first SDP-relaxation in the hierarchy to solve problem $\mathbf{P}$, i.e.,

$$\mathbf{Q}_1 : \quad \left\{ \begin{array}{ll} \displaystyle\max_{\mathbf{x},\mathbf{X}} & \mathrm{trace}\,(Q\,\mathbf{X}) \\ \mathrm{s.t.} & \left[\begin{array}{cc} 1 & \mathbf{x}' \\ \mathbf{x} & \mathbf{X} \end{array}\right] \succeq 0, \quad \mathbf{X}' = \mathbf{X} \in \mathbb{R}^{n\times n} \\ & \\ & X_{ii} = x_i,\ i = 1,\ldots,n \\ & \sum_{\ell=1}^n x_i = k \\ & k\,x_i - \sum_{\ell=1}^n X_{i\ell}\ =\ 0, \quad i = 1,\ldots,n \end{array}\right.$$

whereas $\mathbf{P}_1$ denote the cost of the solution $\mathbf{x} \in \{0,1\}^n$ generated by the (J+M)-heuristic.

We have tested the (J+M)-algorithm on problems randomly generated as for MAXCUT, and with $k = n/2 = 10$. The average relative error $\mathbf{Q}_1 - \mathbf{P}_1|/|\mathbf{Q}_1|$ was

- 5.7% on 4 randomly generated problems with $n = 60$ variables,
- 4.5% and 5.6% on 2 randomly generated problems with $n = 70$ variables. The "max-gap" variant was a little better ($\approx 4\%$ and $\approx 4.5\%$ respectively).
- 5.7% on a problem with $n = 80$ variables.

The CPU times were of the same order of magnitude as for the MAXCUT problem.

4.5.2.3 The 0/1 KNAPSACK problem

Finally, we have also tested the (J+M)-algorithm for the 0/1 KNAPSACK problem:

$$\mathbf{P} : \qquad \max_{\mathbf{x}} \left\{ \mathbf{c}'\mathbf{x} \,:\, \mathbf{x} \in \{0,1\}^n;\ \sum_{\ell=1}^n a_\ell\, x_\ell\ \leq\ b \right\}, \tag{4.25}$$

for some real vector $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{a} \in \mathbb{N}^n$, $b \in \mathbb{N}$.

As for the k-CLUSTER problem, the constraint $\sum_\ell a_\ell\, x_\ell \leq b$ is linear. Therefore, again, to take full advantage of the i-th SDP relaxation (4.20) which contains moments z_α of order up to $2i$, one may add the n redundant constraints $x_i(b - \sum_\ell a_\ell x_\ell) \geq 0$, and $(1-x_i)(b-\sum_\ell a_\ell x_\ell) \geq 0$, $i=1,\ldots,n$, in the definition (4.25) of $\mathbf{P}$; they also correspond to the first-level RLT constraints in [70].

Again, and as for MAXCUT and k-CLUSTER problems, $\mathbf{Q}_1$ denotes the optimal value of the first SDP-relaxation in the hierarchy to solve problem $\mathbf{P}$, i.e.,

$$\mathbf{Q}_1 : \left\{ \begin{array}{ll} \max\limits_{\mathbf{x},\mathbf{X}} & \mathbf{c}'\mathbf{x} \\ \text{s.t.} & \left[\begin{array}{cc} 1 & \mathbf{x}' \\ \mathbf{x} & \mathbf{X} \end{array}\right] \succeq 0, \quad \mathbf{X}' = \mathbf{X} \in \mathbb{R}^{n\times n} \\ & \\ & X_{ii} = x_i, \; i=1,\ldots,n \\ & b\,x_i - \sum_{\ell=1}^n a_\ell X_{i\ell} \geq 0, \quad i=1,\ldots,n \\ & b - b\,x_i - \sum_{\ell=1}^n a_\ell (x_\ell - X_{i\ell}) \geq 0, \quad i=1,\ldots,n \end{array} \right.$$

whereas $\mathbf{P}_1$ denote the cost of the solution $\mathbf{x} \in \{0,1\}^n$ generated by the (J+M)-algorithm.

We have tested the “joint+marginal" algorithm on a sample of 16 problems with $n=50$ variables and 3 problems with $n=60$ variables where, $b=\sum_\ell a_\ell/2$, and the integers a_ℓ's are generated uniformly in $[10,100]$. The vector $\mathbf{c}$ is generated by: $c_\ell = s\epsilon + a_\ell$ with $s=0.1$ and ϵ is a random variable uniformly distributed in $[0,1]$. From the results reported in Table 2 (of the paper “A “joint + marginal" heuristic for 0/1 programs" in Appendix C), one may see that very good relative errors are obtained, in accordance with the fact that the 0/1 knapsack problem is considered as an easy problem since there is a Fully Polynomial Time Approximation Scheme. Moreover, recently, Karlin et al. [34] have shown that solving the t-th SDP-relaxation of the Lasserre hierarchy produces an integrality gap of at most $t/(t-1)$. The CPU time is very comparable (for same value of n) to the one for MAXCUT and k-cluster problems since we solve the same sequence of very similar semidefinite programs.

n	50	60
$(\mathbf{Q}_1 - \mathbf{P}_1)/\lvert\mathbf{Q}_1\rvert$	2.1%	0.62%

Table 4.4: Relative error for 0/1 knapsack

Conclusion

Our first preliminary results are promising, even with the smallest possible relaxation order i. In general, the efficiency of **ALGO 1** or **ALGO 2** with i fixed, should be related to how close to the global optimum f^* is the optimal value f_i^* at step i of the hierarchy of relaxations defined in Chapter 2 to approximate the optimal value f^* of the original problem. When the feasible set is non convex, it may become difficult to obtain a feasible solution and an interesting issue for further investigation is how to proceed when $\mathbf{K}_y = \emptyset$ for y in some open subinterval of $\mathbf{Y}_k$ (proceeding by dichotomy on $\mathbf{Y}_k$ is one possiblity).

4.6 The paper details

4.6.1 Continuous cases:

See Appendix B: *A "Joint+marginal" algorithm for polynomial optimization* appeared in Decision and Control (CDC), 2010 49th IEEE Conference. Publication Year: 2010 , Page(s): 3871 - 3876.

4.6.2 Discrete cases:

See Appendix C: *A "joint + marginal" heuristic for 0/1 programs* appeared in Journal of Global Optimization (27 September 2011), pp. 1-16, doi:10.1007/s10898-011-9788-9

Conclusion

The *moment-SOS* approach which consists of solving a hierarchy of semidefinite relaxations of increasing size is a powerful methodology for solving global optimization problems where the objective function is a polynomial and the feasible set is a compact basic semi-algebraic set. However, in view of present status of semidefinite solvers this approach is so far limited to problems of modest size only, unless some symmetries and/or structured sparsity are present in the problem data and can be exploited.

In general, for medium to large size problems only a few (if not only one) relaxations of the hierarchy can be implemented and provide only a lower bound on the global optimum. Therefore the issue addressed in this thesis was *how to use an optimal solution of such relaxations (or a slightly modified version of them) to construct a "good" feasible solution of the original problem.* We have addressed this issue in two different ways:

- In a first contribution we have considered how to use the moment-SOS approach to help solve (potentially) large scale MINLP problems. Typically, to solve (or help solve) MINLP problems, a popular approach is to use Branch & Bound techniques and for obvious efficiency reasons, the lower bounds computed at each node of the search tree need be computed efficiently. To do so, one usually solves a convex relaxation obtained from *convex underestimators* (on some appropriate box) of the objective function and of the polynomials that define non convex constraints. We have thus considered the generic problem of computing a *polynomial* convex underestimator of a given polynomial f on a box $\mathbf{B} = \prod_{i=1}^{n}[a_i, b_i]$ of $\mathbb{R}^n$. The novelty with previous approaches is that the moment-SOS approach permits to minimize directly the important L^1-norm *tightness criterion* $\int_{\mathbf{B}} |f - h| d\mathbf{x}$ (which measures the "error" between f and its convex underestimate h) over all polynomials of degree at most d, convex on $\mathbf{B}$. Indeed one may use SOS-type certificates of increasing quality (and size) to ensure that $h \in \mathbb{R}[\mathbf{x}]_d$ is convex and is dominated by f on $\mathbf{B}$. As a result one obtains a hierarchy of semidefinite programs whose size is parametrized by the size of the latter certificates, and each semidefinite relaxation provides a polynomial convex underestimator of degree d. Experimental computational results have demonstrated that the resulting estimator is significantly better than the popular $\alpha\mathbf{BB}$ underestimator (and some of its refinements), both for the tightness criterion and for the quality of the lower bounds obtained by minimizing the convex underestimator on $\mathbf{B}$.

- In a second contribution we have considered polynomial optimization problems for which only a few semidefinite relaxations of the *moment-SOS* approach can be implemented. The basic idea is to consider the variable x_1 as a *parameter* in some interval $\mathbf{Y}_1 \subset \mathbb{R}$ and compute an approximation of the optimal value function $J(y) = \min\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}; x_1 = y\}$. Then the global optimum satisfies $J^* = \min\{J(y) : y \in \mathbf{Y}_1\}$, a univariate problem. By using results from polynomial parametric optimization, a good polynomial approximation $p_k \in \mathbb{R}[\mathbf{x}]_k$ of $J(y)$ on $\mathbf{Y}_1$, can be obtained by the moment-SOS approach (in solving an appropriate hierarchy of semidefinite programs). In fact $p_k \to J$ as $k \to \infty$ for the L^1-norm on $\mathbf{Y}_1$. But in practice one only computes p_k for the largest possible index k (according to the maximum size of the semidefinite relaxations that can be handled). Then one obtains $\tilde{x}_1 \in \mathbf{Y}_1$ by minimizing the univariate polynomial p_k on $\mathbf{Y}_1$, an easy convex optimization problem. The process is iterated with x_2 instead of x_1, etc., until a point $\tilde{\mathbf{x}} \in \mathbb{R}^n$ is obtained. If the feasible set $\mathbf{K}$ is convex and also for some 0/1 optimization problems, one may use a variant to guarantee that $\tilde{\mathbf{x}}$ is feasible. Computational experiments on a sample of non trivial problems show promising results, and particularly for discrete 0/1 problems like MAXCUT, k-CLUSTER, and KNAPSACK.

Future works

Here are some possible extensions to the work presented in this thesis:

- For convex underestimator problems, it would be interesting to compare the one obtained here by minimizing the L^1-norm and the one that minimizes the sup-norm on $\mathbf{B}$, i.e., $\sup_{\mathbf{x}\in\mathbf{B}} |f(\mathbf{x}) - h(\mathbf{x})|$. The resulting convex underestimator might provide a better lower bound when minimized on $\mathbf{B}$.

- The “joint+marginal" approach for parametric optimization could also be used in *multi-objective optimization* to approximate the Pareto curve. To illustrate the idea, suppose that one has two conflicting polynomial criteria $f_1, f_2 \in \mathbb{R}[\mathbf{x}]$ to minimize. One may obtain a subset $(f_1^*(\lambda), f_2^*(\lambda))$, $\lambda \in [0, 1]$, of the Pareto curve by minimizing $J(\lambda) = \min\{\lambda f_1(\mathbf{x}) + (1-\lambda) f_2(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K}\}$, with $\lambda \in [0, 1]$, a parametric polynomial optimization problem. By using ideas from parametric polynomial optimization, one may obtain (approximate) moments of the functions $f_1^*(\lambda)$ and $f_2^*(\lambda)$ on $[0, 1]$. It remains to approximate the functions $f_1^*(\lambda)$ and $f_2^*(\lambda)$ on $[0, 1]$, which can be done by several methods (like e.g. maximum-entropy).

[]

Bibliography

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, *A Global Optimization method, αBB, for General Twice-Differentiable Constrained NLPs: II - Implementation and Computational Results*, Computers and Chemical Engineering **22** (1998), 1159–1179.

[2] ———, *Global Optimization of Mixed integer nonlinear problems*, AIChE Jour **46** (2000), 1769–1797.

[3] C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier, *A Global Optimization method, αBB, for General Twice-Differentiable Constrained NLPs - I. Theoretical advances*, Computers and Chemical Engineering **22** (1998), 1137–1158.

[4] I. G. Akrotirianakis and C. A. Floudas, *A new class of improved convex underestimators for twice continuously differentiable constrained NLPs*, J. Global Optim. **30** (2004), 367–390.

[5] F.A. Al-Khayyal and J. E. Falk, *Jointly constrained biconvex programming*, Mathematics of Operations Research **8** (1983), 273–286.

[6] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, αBB *: A global optimization method for general constrained nonconvex problems*, J. Global Optim **7(4)** (1995), 337–363.

[7] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wachter, *Branching and bounds tightening techniques for non-convex minlp*, Opt. Meth. and Software **24** (2009), 597–634.

[8] G. Blekherman, *There are significantly more nonnegative polynomials than sums of squares*, Israel J. Math. **153** (2006), 355–380.

[9] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuejols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, and N. Sawaya, *An algorithmic framework for convex mixed integer nonlinear programs*, Discr. Opt. **5** (2008), 186–204.

[10] S. Burer and A. N. Letchford, *Non-Convex Mixed-Integer nonlinear programming: A survey*, Submited to Surveys in Operations Research and Management Science.

[11] S. Cafieri, J. Lee, and L. Liberti, *On convex relaxations of quadrilinear terms*, J. Global Optim **47** (2010), 661–685.

[12] Fair Isaac Corporation, *Xpress-mp reference*, Manual.

[13] R. Curto and L. A. Fialkow, *The truncated complex k-moment problem*, Transactions of the American Mathematical Society **352** (2000), no. 6, pp. 2825–2855.

[14] W. F. de la Vega and C. Kenyon-Mathieu, *Linear programming relaxations of maxcut*, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2007, pp. 53–61.

[15] C. A. Floudas et al., *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1999.

[16] O. Exler, L. T. Antelo, J. A. Egea, A. A. Alonso, and J. R. Banga, *A tabu search-based algorithm for mixed-integer nonlinear problems and its application to integrated process and control system design*, Comput. and Chem. Eng. **32** (2008), 1877–1891.

[17] C. A. Floudas, *Deterministic Global Optimization: Theory, Methods and Applications*, Kluwer Academic Publishers, 2000.

[18] C. A. Floudas and P. Pardalos (Eds.), *Encyclopedia of Optimization*, Kluwer Academic Publishers, Dordrecht, 2001.

[19] K. Gaterman and P. Parrilo, *Symmetry group, semidefinite programs and sums of squares*, J. Pure Appl. Alg. **192** (2004), 95–128.

[20] M. X. Goemans and D. P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, Journal of the ACM **42** (1995), 1115–1145.

[21] G. Golub and C. V. Loan, *Matrix Computations*, The John Hopkins University Press, 3rd edition, New York, 1996.

[22] C. E. Gounaris and C. A. Floudas, *Tight convex underestimators for $\mathtt{C}^2$-continuous problems: I. Univariate functions*, J. Global Optim. **42** (2008), 51–67.

[23] ———, *Tight convex underestimators for $\mathtt{C}^2$-continuous problems: II. Multivariate functions*, J. Global Optim. **42** (2008), 69–89.

[24] O. K. Gupta and V. Ravindran, *Branch and bound experiments in convex nonlinear integer programming*, Mngt. Sci **31** (1985), 1533–1546.

[25] D. Handelman, *Representing polynomials by positive linear functions on compact convex polyhedra*, Pac. J. Math. **132** (1988), 35–62.

[26] C. Helmberg., *Semidefinite Programming for Combinatorial Optimization*, Habilitationsschrift, TU Berlin, 2000.

[27] R. Hemmecke, M. Koppe, J. Lee, and R. Weismantel, *50 Years of Integer Programming 1958-2008*, Nonlinear Integer Programming, Berlin: Springer, 2010, pp. 561–618.

[28] D. Henrion and J. B. Lasserre, *Detecting global optimality and extracting solutions in gloptipoly*, Chapter in D. Henrion, A. Garulli (Editors). Positive polynomials in control. Lecture Notes in Control and Information Sciences, Springer Verlag, 2005.

[29] D. Henrion, J. B. Lasserre, and J. Lofberg, *Gloptipoly 3: Moments, optimization and semidefinite programming*, Optim. Methods and Softwares **24** (2009), 761–779.

[30] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Berlin: Springer-Verlag, 1993.

[31] K. Howard, *How good is the Goemans - Williamson max cut algorithm ?*, SIAM J. COMPUT. **29(1)** (1999), 336–350.

[32] IBM, *Using the cplex callable library*, Version 12.

[33] S. Kale and C. Seshadhri, *Combinatorial approximation algorithms for maxcut using random walks*, CoRR (2010), 1008.2938.

[34] S. Karlin and W. Studden, *Tchebycheff Systems: With Applications in Analysis and Statistics*, John Wiley and Sons, New York, 1966.

[35] S. Kim, M. Kojima, and H. Waki, *Exploiting sparsity in SDP relaxation for sensor network localization*, SIAM J. Optim. **20** (2009), 192–215.

[36] G. R. Kocis and I. E. Grossmann, *Computational experience with dicopt solving minlp problems in process synthesis*, Comput. and Chem. Eng. **13** (1989), 307–315.

[37] M. Kojima, S. Kim, and M. Maramatsu, *Sparsity in sums of squares of squares of polynomials*, Math. Program. **11** (2007), 315–336.

[38] W. Kramer and I. Geulig, *Interval Calculus in Maple*, Bergische Universitat, GH Wuppertal., 2001.

[39] J. L. Krivine, *Anneaux preordonnes*, J. Anal. Math. **12** (1964a), 307–326.

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

[41] ______, *Polynomial programming: LP-relaxations also converge*, SIAM J. Optim. **15** (2004), 383–393.

[42] ______, *Convergent SDP-relaxations in polynomial optimization with sparsity*, SIAM J. Optim. **17** (2006), 822–843.

[43] ______, *A semidefinite programming approach to the generalized problem of moments*, Mathematical Programming **112** (2008), 65–92.

[44] ______, *Moments, Positive Polynomials and their Applications*, Imperial College Press, 2009.

[45] ______, *A "joint+marginal" approach to parametric polynomial optimization*, SIAM J. Optim. **20** (2010), 1995–2022.

[46] M. Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0/1 programming*, Math. Oper. Res. **28** (2003), 470–496.

[47] ______, *Sums of squares, moment matrices and optimization over polynomials*, Emerging Applications of Algebraic Geometry, Springer, New York, 2008, pp. 157–270.

[48] J. Lee and S. Leyffer, *Mixed Interger Nonlinear Programming*, Springer, 2012.

[49] L. Liberti, *Comparison of convex relaxations for monomials of odd degree*, Optimization and optimal control **1** (2003), 165–174.

[50] L. Liberti and C. C. Pantelides, *Convex envelopes of monomials of odd degree*, J. Global Optim **25(2)** (2003), 157–168.

[51] Y. Lin and L. Schrage, *The global solver in the lindo api*, Opt. Methods and Software **24** (2009), 657–668.

[52] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and 0-1 optimization problems*, SIAM J. Optim. **1** (1991), 166–190.

[53] C. D. Maranas and C. A. Floudas, *Global minimum potential energy conformations of small molecules*, J. Global Optim **4** (1994), 135–170.

[54] ———, *Finding all solutions of nonlinearly constrained systems of equations*, J. Global Optim **7(2)** (1995), 143–182.

[55] C. A. Meyer and C. A. Floudas, *Convex hull of trilinear monomials with mixed sign domains*, J. Global Optim. **29(2)** (2004), 125–155.

[56] ———, *Convex envelopes for edge-concave functions*, Mathematical Programming **103** (2005), 207–224.

[57] ———, *Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: Spline* αBB *underestimators*, J. Global Optim. **32** (2005), 221–258.

[58] A. Neumaier, *Encyclopedia of Mathematics and its Applications*, ch. Interval methods for systems of equations, Cambrigde University Press, 1990.

[59] J. Nie and M. Schweighofer, *On the complexity of putinar's positivestellensatz*, J. Complexity **23** (2007), 135–150.

[60] I. Nowak and Vigerske, *Lago: a (heuristic) branch and cut algorithm for nonconvex minlps*, Central Eur. J. Oper. Res. **16** (2008), 127–138.

[61] P. Parrilo, *Structured semidefinite programs and semi-algebraic geometry methods in robustness and optimization*, Ph.D. thesis, California Institute of Technology, 2000.

[62] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana Univ. Math. J. **42** (1993), 969–984.

[63] I. Quesada and I. E. Grossmann, *An lp/nlp based branch and bound algorithm for convex minlp optimization problems*, Comput. and Chem. Eng. **16** (1992), 937–947.

[64] H. Ratschek and J. Rokne, *New Computer Methods for Global Optimization*, Halsted Press, Chichester Great Britain, 1988.

[65] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Program. **77** (1997), 273–299.

[66] H. S. Ryoo and N. V. Sahinidis, *Global optimization of nonconvex nlps and minlps with applications in process design*, Comput. and Chem. Eng. **19** (1995), 551–566.

[67] ———, *Analysis of bounds for multilinear functions*, J. Glob. Optim. **19** (2001), 403–424.

[68] K. Schmüdgen, *The k-moment problem for compact semi-algebraic sets*, Math. Ann. **289(2)** (1991), 203–206.

[69] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester, 1986.

[70] H. Sherali and W. P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM J. Discr. Math. **3** (1990), 411–430.

[71] G. Stengle, *A nullstellensatz and a Positivestellensatz in semi-algebraic geometry*, Math, Ann. **207** (1974), 87–97.

[72] M. Tawarmalani and N. Sahinidis, *Semidefinite relaxations of fractional programs via novel convexification techniques*, J. Global Optim **20** (2001), 137–158.

[73] M. Tawarmalani and N. V. Sahinidis, *Convex extensions and envelopes of lower semi-continuous functions*, Math. Program. **93** (2002), 247–263.

[74] F. Vallentin, *Symmetry in semidefinite programs*, Linear Alg. Appl. **430** (2009), 360–369.

[75] W. R. Wade, *The bounded convergence theorem*, Amer. Math. Monthly **81** (1974), 387–389.

[76] H. Waki, S. Kim, M. Kojima, and M. Maramatsu, *Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity*, SIAM J. Optim. **17** (2006), 218–242.

[77] H. Waki, S. Kim, M. Kojima, M. Maramatsu, and H. Sugimoto, *Sparsepop: a sparse semidefinite programming relaxations of polynomial optimization problems*, ACM Trans. Math. Software. **35** (2008), 15.

[78] T. Westerlund and F. Pettersson, *A cutting plane method for solving convex minlp problems*, Comput. and Chem. Eng. **19** (1995), 131– 136.

[79] L. A. Wolsey, *Integer Programming*, John Wiley and Sons, New York, 1998.

Appendix A

Paper: “Convex underestimators of polynomials"

J Glob Optim
DOI 10.1007/s10898-012-9974-4

Convex underestimators of polynomials

Jean B. Lasserre · Tung Phan Thanh

Received: 4 July 2011 / Accepted: 22 August 2012

Abstract Convex underestimators of a polynomial on a box. Given a non convex polynomial $f \in \mathbb{R}[\mathbf{x}]$ and a box $\mathbf{B} \subset \mathbb{R}^n$, we construct a sequence of convex polynomials $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]$, which converges in a strong sense to the "best" (convex and degree-d) polynomial underestimator f_d^* of f. Indeed, f_d^* minimizes the L_1-norm $\|f - g\|_1$ on $\mathbf{B}$, over all convex degree-d polynomial underestimators g of f. On a sample of problems with non convex f, we then compare the lower bounds obtained by minimizing the convex underestimator of f computed as above and computed via the popular α**BB** method and some of its other refinements. In most of all examples we obtain significantly better results even with the smallest value of k.

Keywords Convex underestimators · Polynomials · Semidefinite programming

1 Introduction

Consider the general polynomial optimization problem **P**:

$$\mathbf{P}: \quad f^* = \min_{\mathbf{x}} \quad f(\mathbf{x}) \\ \text{s.t} \quad g_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, m \\ \mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n.$$

where f, g_i are polynomials and $\mathbf{x}^L, \mathbf{x}^U \in \mathbb{R}^n$ define the box $[\mathbf{x}^L, \mathbf{x}^U] \subset \mathbb{R}^n$. To approximate f^* and global minimizers of **P**, one of the popular methods (especially for large scale optimization problems) is the deterministic global optimization algorithm α**BB**. It uses a branch

J. B. Lasserre ( )
LAAS-CNRS and Institute of Mathematics, University of Toulouse,
7 Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France
e-mail: lasserre@laas.fr

T. P. Thanh
University of Toulouse,
7 Avenue du Colonel Roche, 31077 Toulouse Cedex 4, France
e-mail: tphanta@laas.fr

Published online: 04 September 2012

and bound scheme where the lower bounds computed at nodes of the search tree are obtained by solving a *convex* problem where f is replaced with some convex underestimators on a box $\mathbf{B} \subset \mathbb{R}^n$; see e.g. [6, pp. 1–12], Floudas [5], Androulakis et al. [1]. Of course, the overall efficiency of the α**BB** algorithm depends heavily on the quality of the lower bounds computed in the branch and bound search tree, and so, ultimately, on the quality of the underestimators of f that are used.

Therefore, the development of tight convex underestimators for non convex polynomials on the feasible region (compact or non compact) is of crucial importance. Several results are available in the literature for computing convex envelopes of specific simple functions in *explicit* form, on a box $\mathbf{B} \subset \mathbb{R}^n$. See for instance Floudas [5] for convex envelopes of bilinear, trilinear, multilinear monomials, Tawarmalani and Sahinidis [15]) for fractional terms, and Cafieri et al. [3] for quadrilinear terms. For a general non convex function f, a convex underestimator can be obtained from the original function f by adding a negative part. For instance, this part could be a negative quadratic polynomial of the form

$$\mathbf{x} \mapsto \mathcal{L}(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i \left(x_i - x_i^L\right)\left(x_i - x_i^U\right),$$

e.g., as in Androulakis et al. [1], or an exponential term from the original function of the form

$$\mathbf{x} \mapsto \mathcal{L}(\mathbf{x}) = f(\mathbf{x}) - \sum_{i=1}^{n} \left(1 - \mathrm{e}^{\alpha_i (x_i - x_i^L)}\right)\left(1 - \mathrm{e}^{\alpha_i (x_i^U - x_i)}\right),$$

e.g., as in Akrotirianakis and Floudas [2]). This is the spirit of the α**BB** method, and several heuristics have been proposed for choosing appropriate nonnegative coefficients $\alpha \in \mathbb{R}^n$ in a tradeoff between two conflicting criteria. On the one hand, the additional term must be *negative enough* to overpower all the non convexities, which requires positive semidefiniteness of the Hessian matrix $\nabla^2\mathcal{L}$ of the twice-differentiable function $\mathcal{L}$. But on the other hand, this additional part should also be as small as possible to obtain good lower bounds when using $\mathcal{L}$ as a substitute for f in the Branch and Bound search tree. Indeed, bad lower bounds would slow down convergence of the α**BB** method. The so-called scaled Gershgorin method is among the most efficient.

Finally, to improve the basic α**BB** method Meyer and Floudas [13] have proposed the *spline* α**BB** variant where the resulting convex underestimator is a piecewise quadratic perturbation of the function f. The initial box $\mathbf{B}$ is partitioned into smaller boxes $\mathbf{B}_k$, $k = 1, \ldots, s$, and in each box $\mathbf{B}_k$ a convex underestimator f_k of f on $\mathbf{B}_k$ is computed as in the basic α**BB** method. In addition, for smoothness and continuity, all convex underestimators f_k are constrained to agree on boundaries of boxes (whence the *spline* name). More recently, Gounaris and Floudas [8] have developed another variant of the α**BB** method in which each convex underestimator f_k on $\mathbf{B}_k$ is computed independently of the others f_j's. Then a global convex underestimator on $\mathbf{B}$ is constructed from the f_k's. Both variants have resulted in significant improvements over the basic α**BB** method.

Contribution. We present a new class of convex underestimators for a non convex polynomial on a box $\mathbf{B} \subset \mathbb{R}^n$, different in spirit from the α**BB** convex underestimators. We use two certificates for (a) $\mathcal{L} \leq f$ and (b), convexity of $\mathcal{L}$ on the box $\mathbf{B}$. More precisely, we are looking for a convex polynomial $f_d \in \mathbb{R}[\mathbf{x}]_d$ (with degree d fixed) which approximates f from below on a given box $\mathbf{B} \subset \mathbb{R}^n$. Hence a polynomial candidate f_d must satisfy two major conditions:

- $f \geq f_d$ on $\mathbf{B}$,
- The Hessian matrix $\nabla^2 f_d$ must be positive semidefinite (i.e., $\nabla^2 f_d \succeq 0$) on $\mathbf{B}$.

But of course, there are many potential polynomial candidates $f_d \in \mathbb{R}[\mathbf{x}]_d$ and therefore, a meaningful criterion to select the "best" among them is essential. A natural candidate criterion to evaluate how good is f_d, is the integral $J(f_d) := \int_{\mathbf{B}} |f - f_d| d\mathbf{x}$, which evaluates the L_1-norm of $f - f_d$ on $\mathbf{B}$, i.e., the "tightness" of $f - f_d$. Indeed, minimizing J tends to minimize the discrepancy (or "error") between f and f_d, uniformly on $\mathbf{B}$. If desired, some flexibility is permitted by allowing any weight function $W : \mathbf{B} \to \mathbb{R}$, positive on $\mathbf{B}$, so as to minimize $J_W(f_d) = \int_{\mathbf{B}} |f - f_d| W d\mathbf{x}$.

Fortunately, to certify $f - f_d \geq 0$ and $\nabla^2 f_d \succeq 0$ on $\mathbf{B}$, a powerful tool is available, namely Putinar's Positivstellensatz (or algebraic positivity certificate) [14], already extensively used in many other contexts, and notably in global polynomial optimization; see e.g. [11,12] and the many references therein. Moreover, since $f \geq f_d$, the criterion $J(f_d)$ to minimize becomes $\int_{\mathbf{B}} (f - f_d) d\mathbf{x}$ and is *linear* in the coefficients of f_d! Therefore, we end up with a hierarchy of semidefinite programs, parameterized by some integer $k \in \mathbb{N}$. This parameter k reflects the size (or complexity) of Putinar's positivity certificate. Any optimal solution of a semidefinite program in this hierarchy provides a convex degree-d polynomial underestimator $f_{dk} \in \mathbb{R}[\mathbf{x}]$.

We then provide a sequence of convex degree-d polynomial underestimators $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]_d, k \in \mathbb{N}$, such that $\| f - f_{dk} \|_1 \to \| f - f_d^* \|_1$ for the L_1-norm on $\mathbf{B}$, where f_d^* minimizes $J(h)$ over all convex degree-d polynomial underestimators h of f on $\mathbf{B}$. In fact, any accumulation point φ^* of the sequence $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]_d$ also minimizes $J(h)$ and $f_{dk_i} \to \varphi^*$ pointwise for some subsequence.

So when k increases the convex polynomial underestimator f_{dk} converges to the best convex polynomial underestimator of degree d. However this convergence is only theoretical because in practice one does not let k increase; one let k fixed (and even to a small value). We also prove that if k is sufficiently large, then f_{dk} is necessarily better than the α**BB** underestimator. Finally, a practical justification is also obtained from a comparison with the α**BB** method carried out on a set of test examples taken from the literature [4,5,8]. Recall that the main motivation for computing underestimators is to compute "good" lower bounds on a box $\mathbf{B}$ for non convex problems, and use these lower bounds in a Branch and Bound algorithm. Therefore, to compare the two underestimators,

- we have computed the lower bound obtained by minimizing each one of them (instead of f) on the box $\mathbf{B}$, and
- we have evaluated the same "tightness" criterion $\int_{\mathbf{B}} |f - \hat{f}| d\mathbf{x}$ for both of them. This latter criterion is important because one may also use a convex underestimator $\hat{f}$ to relax a nonconvex constraint $f(\mathbf{x}) \leq 0$ by the convex one $\hat{f}(\mathbf{x}) \leq 0$. The closer $\hat{f}$ is to f, the better is the relaxation.

In all examples, and on both criteria, the results obtained with the moment approach are significantly better. Observe that with respect to the latter criterion, our underestimator is by essence trying to minimize the discrepancy between f and its underestimator $\hat{f}$.

Finally, we also provide an alternative way to compute the coefficients α in the α**BB** method. Namely, we propose to compute the coefficients α which minimize $\int_{\mathbf{B}} |f - \mathcal{L}| d\lambda$ (where $\mathcal{L}$ is the α**BB**-underestimator), which reduces to solving a single semidefinite program. A library of such α could be computed off-line for several important particular cases.

As already mentioned, some refinements combining underestimators defined on boxes of smaller size have been proposed in the literature, notably by Meyer and Floudas [13], Gounaris and Floudas [7,8], with significant improvements over the basic α**BB** method. It is worth emphasize that our convex underestimators can be also implemented in the above refinements [7,8,13]. Namely since in those two variants [8,13] a standard α**BB**-type convex underestimator f_k is computed in each one of the many sub-boxes $\mathbf{B}_k$ of the box $\mathbf{B}$, one may then replace each f_k by a new one $\hat{f}_k$ as proposed in this paper, and then construct a global one as in [8,13]. The rationale behind this strategy is that since our estimator $\hat{f}_k$ is tighter than f_k in each box $\mathbf{B}_k$ (at least in all examples that we have tried), the resulting convex underestimator should also be better in the box $\mathbf{B}$.

At last but not least, we also provide *parametric* convex underestimators on the box $\mathbf{B}_s := [0, s]^n$ where the scalar s parameterizes the box size. The polynomial underestimator is now a degree-d polynomial in $\mathbb{R}[\mathbf{x}, s]$, which is convex on $\mathbf{x}$ for all values of $s \in [0, S]$ (for some fixed S). This can be especially useful in mixed integer non linear programs where in the Branch and Bound search tree, one has to compute lower bounds repeatedly on boxes of various sizes. Then it may be worthy to compute off line a convex underestimator $f_d \in \mathbb{R}[\mathbf{x}, s]$ and then on line in the search tree, one instantiates $\mathbf{x} \mapsto f_d(\mathbf{x}, s) \in \mathbb{R}[\mathbf{x}]_d$ when s is fixed at the value of the desired box size.

Computational burden. Typically in large scale problems (in particular, mixed integer nonlinear programs), the non convex objective function f is a sum of many functions f_i, each with a small number of variables. As convex underestimators of f would be too costly to compute, the common practice is to add up convex underestimators of the f_i's, much easier to obtain and which can be computed separately. Hence the moment approach described in this paper can be implemented. However, if some sparsity is present in the data then it may be worth trying the specific and efficient semidefinite relaxations of Waki et al. [16] that take sparsity into account, to compute a convex underestimator of f. (Such "sparse" semidefinite relaxations have been implemented in [16] for solving some non convex optimization problems with up to a thousand variables!)

2 Notation and definitions

Let $\mathbb{R}[\mathbf{x}]$ be the ring of real polynomials in the n variables $\mathbf{x} = (x_1, \ldots x_n)$, and for every $d \in N$, let $\mathbb{R}[\mathbf{x}]_d \subset \mathbb{R}[\mathbf{x}]$ be the vector space of polynomials of degree at most d whose dimensions is $s(d) := \binom{n+d}{n}$. Similarly, let $\mathbb{R}[\mathbf{x}, \mathbf{y}]_d \subset \mathbb{R}[\mathbf{x}, \mathbf{y}]$ be the vector space of polynomials of degree at most d whose dimension is $v(d) := \binom{2n+d}{2n}$. Also, let $\Sigma[\mathbf{x}]_d \subset \mathbb{R}[\mathbf{x}]$ be the cone of sums of squares of degree at most $2d$. With $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n$, being the canonical (monomial) basis of $\mathbb{R}[\mathbf{x}]$, a polynomial $f \in \mathbb{R}[\mathbf{x}]_d$ is written

$$\mathbf{x} \mapsto f(\mathbf{x}) = \sum_{\alpha \in \mathbb{N}^n} f_\alpha \, \mathbf{x}^\alpha,$$

for some vector of coefficients $\mathbf{f} = (f_\alpha) \in \mathbb{R}^{s(d)}$.

For every $\alpha \in \mathbb{N}^n$ let $|\alpha| := \sum_i \alpha_i$, and let $\mathbb{N}^n_d := \{\alpha \in \mathbb{N}^n : |\alpha| \leq d\}$. Let the box $\mathbf{B} := [0, 1]^n$ be described as the compact basic semi-algebraic set:

$$\mathbf{B} := \left\{ \mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x})(:= x_j(1 - x_j)) \geq 0, \; j = 1, \ldots, n \right\}.$$

Let g_o be the constant polynomial equal to 1, and let $Q_{\mathbf{B}} \subset \mathbb{R}[\mathbf{x}]$ be the quadratic module associated with the $g_j^\prime s$, i.e.,

$$Q_{\mathbf{B}} := \left\{ \sum_{j=0}^{n} \sigma_j g_j : \sigma_j \in \Sigma[\mathbf{x}],\, j = 1, \ldots, n \right\}.$$

The quadratic module $Q_{\mathbf{B}}$ is *Archimedean*, i.e., there exists some $M > 0$ such that the quadratic polynomial $\mathbf{x} \mapsto M - \|\mathbf{x}\|^2$ belongs to $Q_{\mathbf{B}}$. The following result is a direct consequence of Putinar's Positivstellensantz [14] for Archimedean quadratic modules.

Proposition 2.1 (Putinar [14]) *Every polynomial strictly positive on* $\mathbf{B}$ *belongs to* $Q_{\mathbf{B}}$.

Let $\mathbf{K} \in \mathbb{R}^n$ be the closure of some open bounded set, and let $\mathbf{U} := \{\mathbf{x} \in \mathbb{R}^n := \|\mathbf{x}\|^2 \leq 1\}$. Recall that $f \in \mathbb{R}[\mathbf{x}]_d$ is convex on $\mathbf{K}$ if and only if $\nabla^2 f(\mathbf{x})$ is positive semidefinite on $\mathbf{K}$. Equivalently, f is convex if and only if $\mathbf{T} f_d \geq 0$ on $\mathbf{K} \times \mathbf{U}$, where $\mathbf{T} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}[\mathbf{x}, \mathbf{y}]$ is the mapping:

$$h \mapsto \mathbf{T}h(\mathbf{x}, \mathbf{y}) := \mathbf{y}'\nabla^2 h(\mathbf{x})\mathbf{y}, \quad \forall h \in \mathbb{R}[\mathbf{x}]. \tag{1}$$

The vector of coefficients $((\mathbf{T}h)_{\alpha\beta})$, $\alpha, \beta \in \mathbb{N}^n$, of the polynomial $\mathbf{T}h \in \mathbb{R}[\mathbf{x}, \mathbf{y}]$ is a vector with finitely many zeros and is obtained from the vector $\mathbf{h}$ of $h \in \mathbb{R}[\mathbf{x}]$ by a linear mapping with associated infinite matrix $\mathbf{T}$ whose rows (resp. columns) are indexed in the canonical basis of $\mathbb{R}[\mathbf{x}, \mathbf{y}]$ (resp. $\mathbb{R}[\mathbf{x}]$) and with entries:

$$\mathbf{T}((\alpha, \beta), \delta) = (\mathbf{T}\mathbf{x}^{\delta})_{\alpha\beta}, \quad \alpha, \beta, \delta \in \mathbb{N}^n. \tag{2}$$

Next let $\mathbf{f} = (f_\alpha)$ be the vector of coefficients of $f \in \mathbb{R}[\mathbf{x}]$. Expanding the polynomial $\mathbf{T} f = \mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y}$ in the canonical basis $(\mathbf{x}^\alpha \mathbf{y}^\beta)$ of $\mathbb{R}[\mathbf{x}, \mathbf{y}]_d$, yields

$$\mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y} = \sum_{(\alpha,\beta)\in\mathbb{N}_d^{2n}} (\mathbf{T} f)_{\alpha\beta} \mathbf{x}^\alpha \mathbf{y}^\beta = \sum_{\delta\in\mathbb{N}_d^n} f_\delta \mathbf{T}\mathbf{x}^\delta.$$

3 Main result

Let λ denote the Borel probability measure uniformly distributed on the unit ball $\mathbf{B} := [0, 1]^n$ (i.e. a normalization of the Lebesgue measure on $\mathbb{R}^n$), and consider the associated optimization problem:

$$\min_{h\in\mathbb{R}[\mathbf{x}]_d} \left\{ \int_{\mathbf{B}} (f - h) d\lambda : \quad f - h \geq 0 \text{ on } \mathbf{B};\, h \text{ convex on } \mathbf{B} \right\}. \tag{3}$$

whose optimal value is denoted by ρ_d. Equivalently,

$$\rho_d = \min_{h\in\mathbb{R}[\mathbf{x}]_d} \left\{ \int_{\mathbf{B}} (f - h) d\lambda : \quad f - h \geq 0 \text{ on } \mathbf{B};\, \mathbf{T}\, h \geq 0 \text{ on } \mathbf{S} \right\}, \tag{4}$$

where $\mathbf{T}$ is defined in (1), and $\mathbf{S} = \mathbf{B} \times \mathbf{U}$ with $\mathbf{U} := \{\mathbf{y} \in \mathbb{R}^n := \|\mathbf{y}\|^2 \leq 1\}$.

Lemma 3.1 *The optimization problem (4) has an optimal solution* $f_d^* \in \mathbb{R}[\mathbf{x}]_d$*.*

Proof Observe that for every feasible solution $f_d \in \mathbb{R}[\mathbf{x}]_d$, $f_d \leq f$ on $\mathbf{B}$ and so

$$\int_{\mathbf{B}} (f - f_d)\, d\lambda = \int_{\mathbf{B}} |f - f_d|\, d\lambda = \|f - f_d\|_1,$$

where $\|\cdot\|_1$ denotes the norm on $L_1([0,1]^n)$, and which also defines a norm on $\mathbb{R}[\mathbf{x}]_d$ (or, equivalently, on $\mathbb{R}^{s(d)}$). Indeed, if $f, g \in \mathbb{R}[\mathbf{x}]$ and $\|f - g\|_1 = 0$ then $f = g$, almost everywhere on $\mathbf{B}$, and so on all of $\mathbf{B}$ because both are polynomials and $\mathbf{B}$ has nonempty interior. So if $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]_d$, $k \in \mathbb{N}$, is a minimizing sequence then $f_{dk} \in \Delta_a := \{h \,:\, \|f - h\|_1 \leq a\}$ for all k (where $a := \int_{\mathbf{B}} (f - f_{d0})\, d\lambda$), and $\int_{\mathbf{B}} (f - f_{dk})\, d\lambda \to \rho_d$ as $k \to \infty$. Notice that $\Delta_a \subset \mathbb{R}[\mathbf{x}]_d$ is a ball and a compact set. Therefore, there is a subsequence k_i and a element $f_d^* \in \Delta_a$ such that $f_{dk_i} \to f_d^*$ as $i \to \infty$. Therefore, $f_{dk_i}(\mathbf{x}) \to f_d^*(\mathbf{x})$ for every $\mathbf{x} \in \mathbf{B}$. Next, since $\mathbf{B}$ is bounded we also have $\sup_i \|f_{dk_i}\|_\infty < M$ for some M, and as a consequence of the Bounded Convergence Theorem (see e.g. Wade [17]),

$$\rho_d = \lim_{i\to\infty} \int_{\mathbf{B}} (f - f_{dk_i})\, d\lambda \to \int_{\mathbf{B}} (f - f_d^*)\, d\lambda.$$

It remains to prove that f_d^* is a feasible solution of (4). So, let $\mathbf{x} \in \mathbf{B}$ be fixed, arbitrary. Then since $f - f_{dk} \geq 0$ on $\mathbf{B}$, the pointwise convergence $f_{dk_i} \to f_d^*$ yields $f(\mathbf{x}) - f_d^*(\mathbf{x}) \geq 0$. Hence $f - f_d^* \geq 0$ on $\mathbf{B}$. Similarly, let $(\mathbf{x}, \mathbf{y}) \in \mathbf{S}$ be fixed, arbitrary. Again, from $\mathbf{T} f_{dk}(\mathbf{x}, \mathbf{y}) \geq 0$, the convergence $f_{dk_i} \to f_d^*$, and the definition of $\mathbf{T}$ in (1), it immediately follows that $\mathbf{T} f_d^*(\mathbf{x}, \mathbf{y}) \geq 0$. Therefore, $\mathbf{T} f_d^* \geq 0$ on $\mathbf{S}$, and so f_d^* is feasible for (4). □

With $\mathbf{U} := \{\mathbf{y} \in \mathbb{R}^n \,:\, \|\mathbf{y}\|^2 \leq 1\}$, the set $\mathbf{S} = \mathbf{B} \times \mathbf{U} \subset \mathbb{R}^{2n}$ is a compact basic semi-algebraic set. So, let $Q_{\mathbf{S}} \subset \mathbb{R}[\mathbf{x}, \mathbf{y}]$ be the quadratic module associated with $\mathbf{S}$, i.e.,

$$Q_{\mathbf{S}} = \left\{ \sum_{j=0}^{n+1} \theta_j\, g_j \;:\; \theta_j \in \Sigma[\mathbf{x}, \mathbf{y}],\ j = 0, \ldots, n+1 \right\},$$

where $(\mathbf{x}, \mathbf{y}) \mapsto g_{n+1}(\mathbf{x}, \mathbf{y}) := 1 - ||\mathbf{y}||^2$; it is straightforward to show that $Q_{\mathbf{S}}$ is Archimedean.

By Proposition 2.1, $\rho = \int_{\mathbf{B}} f\, d\lambda - \rho_d$, and the optimal solution f_d^* of (3) is an optimal solution of the problem $\mathbf{P}_d$ defined by:

$$\rho_d = \max_{h \in \mathbb{R}[\mathbf{x}]_d} \left\{ \int_{\mathbf{B}} h\, d\lambda \,:\; f - h \in Q_{\mathbf{B}};\;\; \mathbf{T}h \in Q_{\mathbf{S}} \right\}. \tag{5}$$

So with $\mathbf{T}$ being the mapping defined in (1), introduce the following semidefinite relaxation $\mathbf{P}_{dk}$ of $\mathbf{P}_d$, defined by:

$$\begin{cases} \max\limits_{h\in\mathbb{R}[\mathbf{x}]_d,\sigma_j,\theta_\ell} & \int\limits_{\mathbf{B}} h\,d\lambda \\ & f(\mathbf{x}) = h(\mathbf{x}) + \sum\limits_{j=0}^{n} \sigma_j(\mathbf{x})g_j(\mathbf{x}) \quad \forall \mathbf{x} \\ \text{s.t.} & \mathbf{T}h(\mathbf{x},\mathbf{y}) = \sum\limits_{j=0}^{n} \theta_j(\mathbf{x},\mathbf{y})g_j(\mathbf{x}) \\ & \qquad + \theta_{n+1}(\mathbf{x},\mathbf{y})g_{n+1}(\mathbf{x},\mathbf{y}) \quad \forall \mathbf{x},\mathbf{y} \\ & \sigma_0 \in \Sigma[\mathbf{x}]_k,\ \sigma_j \in \Sigma[\mathbf{x}]_{k-1},\ j \geq 1 \\ & \theta_0 \in \Sigma[\mathbf{x},\mathbf{y}]_k, \theta_j \in \Sigma[\mathbf{x},\mathbf{y}]_{k-1},\ j \geq 1, \end{cases} \tag{6}$$

with $k \geq \max[\lceil d/2 \rceil, \lceil (\deg f)/2 \rceil]$ and with optimal value denoted by ρ_{dk}.

(6) is a semidefinite program. To see that (6) is a semidefinite program, recall that $h = \sum_{\alpha\in\mathbb{N}_d^n} h_\alpha\,\mathbf{x}^\alpha$. Moreover, let $\mathbf{v}_k(\mathbf{x}) \in \mathbb{R}^{s(k)}$ be the vector of monomials $(\mathbf{x}^\alpha), \alpha \in \mathbb{N}_k^n$. A polynomial $u \in \mathbb{R}[\mathbf{x}]_{2k}$ is of the form $\sigma_j\,g_j$ with $\sigma_j \in \Sigma[\mathbf{x}]_{k-1}$ if and only if there exists some real symmetric $s(k-1) \times s(k-1)$ matrix $\mathbf{Z}_j \succeq 0$ such that $u_\alpha = \langle \mathbf{Z}_j, \mathbf{C}_\alpha^j \rangle$ for all $\alpha \in \mathbb{N}_{2k}^n$; the matrices $\mathbf{C}_\alpha^j, \alpha \in \mathbb{N}_{2k}^n$, are the coefficients of $\mathbf{x}^\alpha$ in the expansion

$$\mathbf{x} \mapsto g_j(\mathbf{x})\,\mathbf{v}_{k-1}(\mathbf{x})\mathbf{v}_{k-1}(\mathbf{x})' = \sum_{\alpha\in\mathbb{N}_{2k}^n} \mathbf{C}_\alpha^j\,\mathbf{x}^\alpha,$$

Similarly, let $\mathbf{w}_k(\mathbf{x},\mathbf{y})$ be the vector of monomials $(\mathbf{x}^\alpha\mathbf{y}^\beta), (\alpha,\beta) \in \mathbb{N}_k^{2n}$. A polynomial $v \in \mathbb{R}[\mathbf{x},\mathbf{y}]_{2k}$ is of the form $\theta_j\,g_j$ with $\theta_j \in \Sigma[\mathbf{x},\mathbf{y}]_{k-1}$ if and only if there exists some real symmetric $v(k-1) \times v(k-1)$ matrix $\Theta_j \succeq 0$ such that $v_{\alpha\beta} = \langle \Theta_j, \Delta_{\alpha\beta}^j \rangle$ for all $(\alpha,\beta) \in \mathbb{N}_{2k}^{2n}$; the matrices $\Delta_{\alpha\beta}^j, (\alpha,\beta) \in \mathbb{N}_{2k}^{2n}$, are the coefficients of $\mathbf{x}^\alpha\mathbf{y}^\beta$ in the expansion

$$(\mathbf{x},\mathbf{y}) \mapsto g_j(\mathbf{x})\,\mathbf{w}_{k-1}(\mathbf{x},\mathbf{y})\mathbf{w}_{k-1}(\mathbf{x},\mathbf{y})' = \sum_{(\alpha,\beta)\in\mathbb{N}_{2k}^{2n}} \Delta_{\alpha\beta}^j\,\mathbf{x}^\alpha\,\mathbf{y}^\beta.$$

For more details the interested reader is referred to e.g. [12]. Therefore (6) is equivalent to:

$$\begin{cases} \max\limits_{h\in\mathbb{R}[\mathbf{x}]_d,\mathbf{Z}_j,\Theta_\ell} & \sum\limits_{\alpha\in\mathbb{N}_d^n} h_\alpha\gamma_\alpha \\ & f_\alpha = h_\alpha + \sum\limits_{j=0}^{n} \langle \mathbf{Z}^j, \mathbf{C}_\alpha^j \rangle, \qquad \forall\,\alpha \in \mathbb{N}_{2k}^n \\ \text{s.t.} & (\mathbf{T}h)_{\alpha\beta} = \sum\limits_{\ell=0}^{n+1} \langle \Theta_\ell, \Delta_{\alpha\beta}^\ell \rangle, \qquad \forall\,(\alpha,\beta) \in \mathbb{N}_{2k}^{2n} \\ & \mathbf{Z}^j,\ \Theta^\ell \succeq 0, \quad j = 0,\ldots,n;\ \ell = 0,\ldots,n+1, \end{cases} \tag{7}$$

where $f_\alpha = 0$ for all α with $|\alpha| > \deg f$, and $\gamma_\alpha = \int_{\mathbf{B}} \mathbf{x}^\alpha\,d\lambda$ for every $\alpha \in \mathbb{N}^n$.

Lemma 3.2 *For sufficiently large $2k \geq \deg f$, the semidefinite program (7) has an optimal solution $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$.*

The proof being a little technical is postponed to Appendix.

Theorem 3.3 *Let ρ_d be the optimal value of (4) and consider the hierarchy of semidefinite relaxations (6) with associated sequence of optimal values (ρ_{dk}), $k \in \mathbb{N}$. Then $\int_{\mathbf{B}} f d\lambda - \rho_{dk} \downarrow \rho_d$ as $k \to \infty$, so that $\| f - f_{dk}\|_1 \downarrow \rho_d$ if $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$ is any optimal solution of (6). Moreover, any accumulation point $\varphi^* \in \mathbb{R}[\mathbf{x}]_d$ of the sequence $(f_{dk}) \subset \mathbb{R}[\mathbf{x}]_d$, is an optimal solution of (4), and $f_{dk_i} \to \varphi^*$ pointwise for some subsequence (k_i), $i \in \mathbb{N}$.*

Proof Let $f_d^* \in \mathbb{R}[\mathbf{x}]_d$ be an optimal solution of (4), which by Lemma 3.1, is guaranteed to exist. As f_d^* is convex on $\mathbf{B}$, $s\nabla^2 f_d^* \succeq 0$ on $\mathbf{B}$. Let $\epsilon > 0$ be fixed and such that $\epsilon\|\mathbf{x}\|^2 < 1$ on $\mathbf{B}$. Let $g_\epsilon := f_d^* - \epsilon + \epsilon^2\|\mathbf{x}\|^2$, so that $\nabla^2 g_\epsilon \succeq \epsilon^2\mathbf{I}$ on $\mathbf{B}$. Hence, by the matrix version of Putinar's Theorem (see [12, Theorem 2.22]), there exist SOS matrix polynomials $\mathbf{F}_j$ such that

$$\nabla^2 g_\epsilon(\mathbf{x}) = \mathbf{F}_0(\mathbf{x}) + \sum_{j=1}^{n} \mathbf{F}_j(\mathbf{x})\, g_j(\mathbf{x}).$$

(Recall that an SOS matrix polynomial $\mathbf{F} \in \mathbb{R}[\mathbf{x}]^{q\times q}$ is a matrix polynomial of the form $\mathbf{x} \mapsto \mathbf{L}(\mathbf{x})'\mathbf{L}(\mathbf{x})$ where $\mathbf{L}$ is a matrix polynomial $\mathbf{L} \in \mathbb{R}[\mathbf{x}]^{p\times q}$ for some $p \in \mathbb{N}$.)

And so, for every $j = 0, \ldots, n$, the polynomial $(\mathbf{x}, \mathbf{y}) \mapsto \theta_j^\epsilon(\mathbf{x}, \mathbf{y}) := \mathbf{y}'\mathbf{F}_j\mathbf{y}$ is SOS for every $j = 0, \ldots, n+1$, and

$$\mathbf{T}g_\epsilon = \sum_{j=0}^{n} \theta_j^\epsilon(\mathbf{x}, \mathbf{y})\, g_j(\mathbf{x}) + \theta_{n+1}^\epsilon(\mathbf{x}, \mathbf{y})(1 - \|\mathbf{y}\|^2).$$

Moreover, observe that $f - g_\epsilon = f - f_d^* + \epsilon(1 - \epsilon\|\mathbf{x}\|^2)$ is strictly positive on $\mathbf{B}$. Hence by Putinar's Theorem,

$$f - g_\epsilon = \sum_{j=0}^{n} \sigma_j^\epsilon g_j,$$

for some SOS polynomials $\sigma_j \in \mathbb{R}[\mathbf{x}]$, $j = 1, \ldots, n$. Let $2t \geq \max\{[\max_k \deg \sigma_k + 2, \max_j [\deg \mathbf{F}_j + 4]\}$. Then the polynomial g_ϵ is a feasible solution of (6) whenever $k \geq t$. Its value satisfies

$$\int_{\mathbf{B}} g_\epsilon d\lambda = \int_{\mathbf{B}} (f_d^* - \epsilon + \epsilon^2\|\mathbf{x}\|^2) d\lambda \geq \int_{\mathbf{B}} f_d^* d\lambda - \epsilon,$$

and so $\rho_{dt} \geq \rho_d - \epsilon$. As $\epsilon > 0$ was arbitrary and the sequence (ρ_{dk}) is monotone non decreasing, the first result follows.

Next, any optimal solution $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$ of (6) satisfies $\| f - f_{dk}\|_1 \leq \int_{\mathbf{B}} f d\lambda - \rho_{d1} =: a$ and so belongs to the ball $\Delta_a := \{h : \| f - h\|_1 \leq a\}$. Let $\varphi^* \in \Delta_a$ be an arbitrary accumulation point of the sequence (f_{dk}) for some subsequence (k_i), $i \in \mathbb{N}$. Proceeding as in the proof of Lemma 3.1, $f_{dk_i} \to \varphi^*$ pointwise, $f - \varphi^* \geq 0$ and $\nabla^2\varphi^* \succeq 0$ on $\mathbf{B}$. Moreover, by the Bounded Convergence Theorem [17],

$$\rho_d = \lim_{i\to\infty} \rho_{dk_i} = \lim_{i\to\infty} \int_{\mathbf{B}} (f - f_{dk_i}) d\lambda = \int_{\mathbf{B}} (f - \varphi^*) d\lambda,$$

which proves that φ^* is an optimal solution of (4). □

Theorem 3.3 states that the optimal value of the semidefinite relaxation (6) can become as close as desired to that of problem (4), and accumulation points of solutions of (6) are also optimal solutions of (4). The price to pay is the size of the semidefinite program (6) which becomes larger and larger as k increases. In practice one let k fixed at a small value and the computational experiments presented below indicate that even with k fixed at its smallest possible value, the polynomial underestimator f_{dk} provides better lower bounds than the α**BB**-underestimator.

4 Comparing the moment and αBB methods

4.1 Convex underestimators from the α**BB** method

To obtain a convex underestimator of a non convex polynomial, the α**BB** method is based on a decomposition of f into a sum of non convex terms of special type (e.g., linear, bilinear, tri-linear, fractional, fractional tri- and quadri-linear) and non convex terms of arbitrary type. The terms of special type are replaced with their convex envelopes which are already known (see Floudas [5]).

For an arbitrary type f, the underestimator $\mathcal{L}$ is obtained by adding a separable negative quadratic polynomial, i.e.,

$$\mathcal{L}(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i \left(x_i - x_i^L\right)\left(x_i - x_i^U\right), \tag{8}$$

where the positive coefficients α_i's are determined so as to make the polynomial underestimator $\mathcal{L}$ convex. As $\mathcal{L}$ is convex on $\mathbf{B}$ if and only if its Hessian $\nabla^2\mathcal{L}$ is positive semidefinite on $\mathbf{B}$, the coefficients $\alpha_i, , i = 1, \ldots, n$ must satisfy

$$\nabla^2\mathcal{L}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) + 2\Delta \succeq 0, \quad \forall \mathbf{x} \in \mathbf{B}, \tag{9}$$

where $\Delta = \mathrm{diag}\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is referred to as the *diagonal shift matrix*. The *separation distance* between the original polynomial f and its convex underestimator $\mathcal{L}$ is

$$d_{\alpha\mathbf{BB}} = f(\mathbf{x}) - \mathcal{L}(\mathbf{x}) = -\sum_{i=1}^{n} \alpha_i \left(x_i - x_i^L\right)\left(x_i - x_i^U\right) \geq 0,$$

which achieves its maximum at the middle point of the interval $[\mathbf{x}^L, \mathbf{x}^U]$. Therefore,

$$d_{\alpha\mathbf{BB}}^{max} = -\frac{1}{4}\sum_{i=1}^{n} \alpha_i \left(x_i^U - x_i^L\right)^2.$$

hence, the value of $d_{\alpha\mathbf{BB}}$ is proportional to the $\alpha_i' s$ and the size of the domains $[\mathbf{x}^L, \mathbf{x}^U]$. A number of methods to calculate the parameters of the diagonal matrix Δ have been developed using interval analysis (see e.g. Floudas [5], Kramer et al. [10]), and are based on the following result:

Theorem 4.1 *Let* $[\mathcal{H}_f]$ *be a real symmetric interval matrix such that* $\nabla^2 f(\mathbf{x}) \in [\mathcal{H}_f], \forall \mathbf{x} \in [\mathbf{x}^L, \mathbf{x}^U]$. *If* $[\nabla^2_{\mathcal{L}}] := [\mathcal{H}_f] + 2\Delta \succeq 0$ *then* $\mathcal{L}$ *is convex on* $[\mathbf{x}^L, \mathbf{x}^U]$.

Among the most efficient methods is the scaled Gershgorin method where $(\alpha_i) \in \mathbb{R}^n$ is determined by

$$\alpha_i = \max\left\{0, -\frac{1}{2}\left(\underline{f}_{ii} - \sum_{j \neq i} \max\{|\underline{f}_{ii}|, |\overline{f}_{ij}|\}\right)\frac{d_j}{d_i}\right\} \tag{10}$$

where $\underline{f}_{ii}$ and $\overline{f}_{ij}$ are the lower and upper bounds of $\partial^2 f/\partial x_i \partial x_j$ in the interval $[\mathbf{x}^L, \mathbf{x}^U]$ and $d_i, i = 1, 2, \ldots, n$ are some chosen positive parameters. The choice $d_i = u_i^U - u_i^L$ reflects the fact that the underestimator is more sensitive to variables with a wide range than to variables with a small range.

4.2 Comparison with the moment method

Given an arbitrary polynomial $f \in \mathbb{R}[\mathbf{x}]$ and $d \in \mathbb{N}$, one searches for an *ideal* polynomial $f_d^* \in \mathbb{R}[\mathbf{x}]_d$ convex on $\mathbf{B}$, that is an optimal solution of $\mathbf{P}_d$, i.e., f_d^* solves:

$$\rho_d = \max_{h \in \mathbb{R}[\mathbf{x}]_d} \left\{ \int_{\mathbf{B}} h\, d\lambda : \quad f - h \in Q_{\mathbf{B}}; \quad \mathbf{T}h \in Q_{\mathbf{S}} \right\}. \tag{11}$$

(See Lemma 3.1.) In practice, one obtains a convex underestimator $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$ by solving the semidefinite relaxation (6) of $\mathbf{P}_d$ for a small value of k.

We can now compare f_{dk} with the α**BB** underestimator $\mathcal{L}$ in (8), with $x_i^L = 0$ and $x_i^U = 1, i = 1, \ldots n$ (possibly after scaling).

Lemma 4.2 *With f being a non convex polynomial, let $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$ be an optimal solution of (6) and let $\mathcal{L}$ be as in (8). If $\nabla^2 \mathcal{L}(\mathbf{x}) \succ 0$ for all $\mathbf{x} \in \mathbf{B}$ then*

$$\| f - f_{dk} \|_1 \leq \| f - \mathcal{L} \|_1, \tag{12}$$

whenever k is sufficiently large. That is, the convex underestimator $f_{dk} \in \mathbb{R}[\mathbf{x}]_d$ is better than $\mathcal{L}$ when evaluated for the L_1-norm $\int_{\mathbf{B}} |f - g| d\lambda$.

Proof Observe that

$$f(\mathbf{x}) - \mathcal{L}(\mathbf{x}) = \sum_{i=1}^n \underbrace{\alpha_i}_{\sigma_i \in \Sigma[\mathbf{x}]_0} x_i(1 - x_i),$$

that is, the separation distance $d_{\alpha\mathbf{BB}}$ is a very specific element of $Q_{\mathbf{B}}$, where the SOS weights σ_j are the constant polynomials $\alpha_j, j = 1, \ldots, n$.

Moreover, if $\mathbf{T}\mathcal{L} \succ 0$ on $\mathbf{B}$ then by [12, Theorem 2.22]

$$\nabla^2 \mathcal{L}(\mathbf{x}) = \sum_{j=0}^n \mathbf{F}_j(\mathbf{x})\, g_j(\mathbf{x}),$$

for some SOS polynomial matrices $\mathbf{x} \mapsto \mathbf{F}_j(\mathbf{x})$ (i.e., of the form $\mathbf{L}_j(\mathbf{x})\mathbf{L}_j(\mathbf{x})'$ for some matrix polynomials $\mathbf{L}_j$) and so

$$\mathbf{T}\mathcal{L}(\mathbf{x}, \mathbf{y}) = \mathbf{y}'\nabla^2 \mathcal{L}(\mathbf{x})\mathbf{y} = \sum_{j=0}^n \underbrace{(\mathbf{L}_j(\mathbf{x})\mathbf{y})^2}_{\theta_j \in \Sigma[\mathbf{x}, \mathbf{y}]} g_j(\mathbf{x}).$$

Hence $\mathbf{T}\mathcal{L} \in Q_{\mathbf{S}}$ and $\mathcal{L}$ is a feasible solution of (6) as soon as $2k \geq \max_j \deg \mathbf{F}_j + 4$. Therefore, at least for sufficiently large k,

$$\int_{\mathbf{B}} f_{dk} d\lambda \geq \int_{\mathbf{B}} \mathcal{L}\, d\lambda,$$

and so as $f \geq f_{dk}$ and $f \geq \mathcal{L}$ on $\mathbf{B}$, (12) holds. □

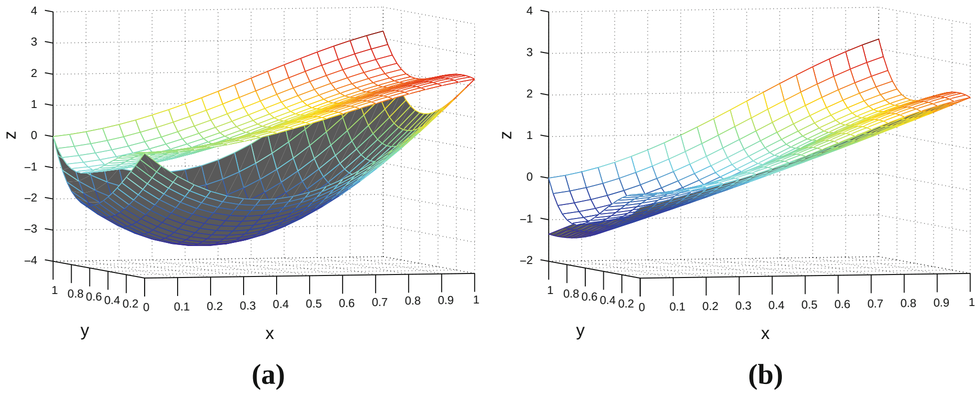


Fig. 1 Bivariate Camelback function. **a** $f^* = -1$; $f^*_{\alpha\mathbf{BB}} = -3.33$, **b** $f^* = -1$; $f^*_{mom} = -1.36$

4.3 Computational results

Among all possible choices of d we consider the two natural choices, $d = \deg f$ or $d = 2$, and $k \geq \max[\lceil d/2 \rceil, \lceil (\deg f)/2 \rceil]$. With the former one searches for the best convex underestimator of same degree as f, while with the latter one searches for the best quadratic underestimator of f.

Recall that the main motivation for computing underestimators is to compute "good" lower bounds on a box $\mathbf{B}$ for non convex problems, and use these lower bounds in a Branch and Bound algorithm. Therefore, to compare the moment and $\alpha\mathbf{BB}$ underestimators, we have chosen non convex optimization problems in the literature, and replaced the original non convex objective function by its moment and $\alpha\mathbf{BB}$ underestimator, respectively f_d and $\mathcal{L}$. We then compare:

- The minimum f^*_{mom} (resp. $f^*_{\alpha\mathbf{BB}}$) obtained by minimizing[1] f_d (resp. $\mathcal{L}$) on the box $\mathbf{B}$.
- The respective values of the "tightness score", i.e., the L_1-norm $\int_{\mathbf{B}} |f - f_d| d\lambda$ and $\int_{\mathbf{B}} |f - \mathcal{L}| d\lambda$. In view of (8), the latter is easy to compute.
  In Fig. 1 is displayed a first illustrative example with the Six-Hump Camelback function (see Meyer [13]) $\mathbf{x} \mapsto f(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$ in the box $\mathbf{B} = [0, 1]^2$. The global minimum is $f^* = -1$ to be compared with $f^*_{mom} = -1.36$, $f^*_{\alpha\mathbf{BB}} = -3.33$.
- Choice 1: $d = 2$ (quadratic underestimator) Given $f \in \mathbb{R}[\mathbf{x}]$, one searches for a convex polynomial $f_d \in \mathbb{R}[\mathbf{x}]_2$ of the form $\mathbf{x} \mapsto f_d(\mathbf{x}) = \mathbf{x}'\mathbf{A}\mathbf{x} + \mathbf{a}'\mathbf{x} + b$ for some real positive semidefinite symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, vector $\mathbf{a} \in \mathbb{R}^n$ and scalar b. Let $\mathbf{M}_\lambda$ be the moment matrix of order 1 of the (normalized) Lebesgue measure λ on $\mathbf{B}$, i.e.,

$$\mathbf{M}_\lambda = \begin{bmatrix} 1 & \gamma' \\ \gamma & \Lambda \end{bmatrix}$$

  with $\gamma_i = \int_{\mathbf{B}} x_i d\lambda$ for all $i = 1, \ldots, n$, and $\Lambda_{ij} = \int_{\mathbf{B}} x_i x_j d\lambda$ for all $1 \leq i, j \leq n$. The semidefinite relaxation $\mathbf{P}_{dk}$ in (6) reads:

[1] All computations were made by running the Gloptipoly software described in Henrion et al. [9], and developed for solving the Generalized Problem of Moments whose global optimization is only a special case. The $\alpha\mathbf{BB}$ underestimator was computed via the scaled Gershgorin method.

Table 1 Comparing f^*_{mom} and $f^*_{\alpha\mathbf{BB}}$; $d = 2$

Prob	n	deg f	d	$[\mathbf{x}^L, \mathbf{x}^U]$	$f^*_{\alpha\mathbf{BB}}$	f^*_{mom}	f^*
Test2	4	3	2	[0,1]	−1.54	−1.22	−1
Test3	5	4	2	[−1,1]	−15	−13.95	−6
Test4	6	6	2	[−1,1]	−60.15	−10.06	−3
Test5	3	6	2	[−2,2]	−411.2	−12.66	−1
Test10	4	4	2	[0,1]	−197.54	−0.9698	0
Test11	4	4	2	[0,1]	−33.02	−0.623	0
Test14(1)	3	4	2	[−5,2]	−2,409	−300	−300
Test14(2)	4	4	2	[−5,2]	−3,212	−400	−400
Test14(3)	5	4	2	[−5,2]	−4,015	−500	−500
Fl.2.2	5	2	2	[0,1]	−18.9	−18.9	−17
Fl.2.3	6	2	2	[0,1]	−5,270.9	−361.50	−361
Fl.2.4	13	2	2	[0,3]	−592	−195	−195
Fl.2.6	10	2	2	[0,1]	−269.83	−269.45	−268.01
Fl.2.8C1	20	2	2	[0,3]	−560	−560	−394.75
Fl.2.8C2	20	2	2	[0,10]	−1,050	−1,050	−884
Fl.2.8C3	20	2	2	[0,30]	−13,600	−12,000	−8,695
Fl.2.8C4	20	2	2	[0,30]	−920	−920	−754.75
Fl.2.8C5	20	2	2	[0,30]	−16,645	−10,010	−4,150.41

$$\begin{cases} \max\limits_{b,\mathbf{a},\mathbf{A}} & b + \mathbf{a}'\gamma + \langle \mathbf{A}, \Lambda \rangle \\ \text{s.t.} & f(\mathbf{x}) = b + \mathbf{a}'\mathbf{x} + \mathbf{x}'\mathbf{A}\mathbf{x} + \displaystyle\sum_{j=0}^{n} \sigma_j(\mathbf{x}) g_j(\mathbf{x}), \qquad \forall \mathbf{x} \\ & \mathbf{A} \succeq 0;\ \sigma_0 \in \Sigma[\mathbf{x}]_k,\ \sigma_j \in \Sigma[\mathbf{x}]_{k-1},\ j \geq 1. \end{cases} \tag{13}$$

Table 1 displays results for some examples and choice $d = 2$. The test functions f are taken from Floudas et al. [4] and Floudas [5] and Gounaris and Floudas [8]. On a box $\mathbf{B}$ that contains the feasible set, we compute the convex $\alpha\mathbf{BB}$ underestimator $\mathcal{L}$ and the (only degree-2) moment underestimator f_d of the initial objective function f via solving (13) with the smallest value of $k = \lceil d/2 \rceil$. We then compute their respective minimum $f^*_{\alpha\mathbf{BB}}$ and f^*_{mom} on $\mathbf{B}$. All examples were run on a Intel(R) Core(TM) i5 2.53 GHz processor with 4 Gb of Ram. In a typical example with degree 4 and 5 variables, the CPU time was 0.68 s to find the underestimator f_d with $d = 2$.

- Choice 2: $d = \deg f$. Table 2 displays results for some examples taken from Table 1 but now using the moment underestimator f_d with choice $d = \deg f$ (the Test 4 example was excluded because f_6 was too expensive to compute). Again k in (6) is set to its smallest possible value $\lceil d/2 \rceil$. As one may see and as expected, the lower bound f^*_{mom} is better, and in several examples f^*_{mom} is very close to the global minimum f^*. However, depending on the degree of f, the computing time is now larger than with $d = 2$; recall that the size of the semidefinite program (6) is parameterized by k, chosen here to be equal to its smallest possible value $\lceil d/2 \rceil$. For a typical example with degree 4 and 5 variables, the CPU time was 1.68 s to find the underestimator f_d with $d = 4$.

Table 2 Comparing f^*_{mom} and $f^*_{\alpha\mathbf{BB}}$; $d = \deg f$

Prob	n	$\deg f$	d	$[\mathbf{x}^L, \mathbf{x}^U]$	$f^*_{\alpha\mathbf{BB}}$	f^*_{mom}	f^*
Test2	4	3	3	[0,1]	-1.54	-1.22	-1
Test3	5	4	4	[-1,1]	-15	-11.95	-6
Test4	6	6	2	[-1,1]	-60.15	-10.06	-3
Test5	3	6	6	[-2,2]	-411.2	-12.07	-1
Test10	4	4	4	[0,1]	-197.54	-0.778	0
Test11	4	4	4	[0,1]	-33.02	0	0
Test14(1)	3	4	4	[-5,2]	$-2{,}409$	-300	-300
Test14(2)	4	4	4	[-5,2]	$-3{,}212$	-400	-400
Test14(3)	5	4	4	[-5,2]	$-4{,}015$	-500	-500

Table 3 Comparing $\frac{\int_{\mathbf{B}}|f-\mathcal{L}|d\lambda}{V_B}$ and $\frac{\int_{\mathbf{B}}|f-f_d|d\lambda}{V_B}$

Prob	$\frac{\int_{\mathbf{B}}|f-\mathcal{L}|d\lambda}{V_{\mathbf{B}}}$	$\frac{\int_{\mathbf{B}}|f-f_d|d\lambda}{V_{\mathbf{B}}}$
Test2	1	0.625
Test3	11.67	3.33
Test4	60	7.29
Test5	99.00	23.20
Test10	133.33	57.00
Test11	46.33	1
Test14(1)	1641.4e+003	149.2711
Test14(2)	2,186.6	199.08
Test14(3)	2,731	248.71
Fl.2.2	41.66	41.66
Fl.2.3	67,500	833.33
Fl.2.4	0.005	0.0005
Fl.2.6	8.33	5.83
Fl.2.8C1	3.4989e+014	3.4989e+014
Fl.2.8C2	12,200	12,200
Fl.2.8C3	6.9979e−005	6.9979e−005
Fl.2.8C4	3.4989e−006	3.4989e−006
Fl.2.8C5	9.6077e−015	9.6077e−015

Finally, Table 3 displays the respective values of the "tightness" score $\int_{\mathbf{B}}|f - \mathcal{L}|d\lambda$ and $\int_{\mathbf{B}}|f - f_d|d\lambda$ with $d = 2$, normalized with respect to the volume of the box **B**. Again, the tightness score of the moment underestimator f_d with $d = 2$ is significantly better than that of the α**BB** underestimator $\mathcal{L}$.

In view of the above experimental results (even though they are limited) it seems that the practical choice $d = 2$ combined with the smallest possible value $k := \lceil(\deg f)/2\rceil$ in (6), is enough to obtain a good convex polynomial underestimator.

4.4 Computing α for the αBB underestimator

The above approach can also be used to provide a new and systematic way to compute the coefficients $\alpha \in \mathbb{R}^n_+$ of the α**BB** underestimator. Indeed it suffices to impose the additional

Table 4 Comparing the $f_{\alpha\mathbf{BB}}$ and f_d^α underestimators

Prob	$[x^L, x^U]$	$f^*_{\alpha\mathbf{BB}}$	f^*_{mom}	f^*
Fl 8.2.7	$[0, 1]^5$	−899.5	−2.76	−0.5
Fl 8.2.7	$[-1, 1]^5$	−2,999	−23	−0.6
Fl 8.2.7	$[-5, 5]^5$	−63,000	−2,987	−982
Test 10	$[0, 1]^4$	−197.5	−61.9	0
Test 10	$[-1, 1]^4$	−870.2	−323.8	0
Test 10	$[-5, 5]^4$	−137e+05	−4.73e+04	−19

requirement that the underestimator (now called f_d^α) has the α**BB** form (8). And so, possibly after a rescaling of the box $\prod_{i=1}^n [x_i^L, x_i^U]$ to $[0, 1]^n$, one wishes to minimize

$$\int_{\mathbf{B}} (f - \mathcal{L}) d\lambda = \underbrace{\int_{\mathbf{B}} f d\lambda}_{\text{constant}} + \sum_{i=1}^n \alpha_i \int_{\mathbf{B}} x_i(1 - x_i) d\lambda, \tag{14}$$

$= \int_{\mathbf{B}} f d\lambda + \frac{1}{6} \sum_{i=1}^n \alpha_i$, under the convexity constraint:

$$\mathbf{y}'\nabla^2\mathcal{L}(\mathbf{x})\mathbf{y} = \sum_{j=1}^{n+1} \theta_j g_j; \quad \theta_j \in \Sigma[\mathbf{x}, \mathbf{y}]_{k-v_j}, \quad j = 0, \ldots, n+1,$$

where $k \geq \lceil(\deg f)/2\rceil$, and $v_j = \lceil(\deg g_j)/2\rceil$, $j = 0, \ldots, n+1$. Therefore computing the best α_i's reduce to solving

$$\begin{aligned} &\min_{\alpha, \theta_j} \sum_{i=1}^n \alpha_i \\ &\text{s.t.} \quad \mathbf{y}'\nabla^2 f(\mathbf{x})\mathbf{y} = -2\sum_{i=1}^n \alpha_i y_i^2 + \sum_{j=0}^{n+1} \theta_j g_j \\ &\qquad \alpha \geq 0; \ \theta_j \in \Sigma[\mathbf{x}, \mathbf{y}]_{k-v_j}, \ j = 0, \ldots, n+1, \end{aligned} \tag{15}$$

which is a semidefinite program. The results displayed in Table 4 for various box sizes and again the smallest value $k = \lceil(\deg f)/2\rceil$ in (15). As can be seen, this strategy can yield significantly better lower bounds than with the scaled Gershgorin method, at least on examples with highly nonconvex functions. Indeed, for various box sizes the resulting lower bound f^*_{mom} obtained by minimizing f_d^α on **B** is always much better than $f^*_{\alpha\mathbf{BB}}$ obtained by minimizing $f_{\alpha\mathbf{BB}}$.

4.5 On other refinements of the αBB method

In the α**BB** method, the size of the interval $d_i = x_i^U - x_i^L$ has a direct impact on the resulting coefficients α_i's and so also on the tightness of the resulting convex underestimator. Therefore some refinements combining underestimators defined on boxes of smaller size have been proposed in the literature. Such refinements, notably by Meyer and Floudas [13], Gounaris and Floudas [7,8], have resulted in significant improvements over the basic α**BB** method.

Basically, the idea is to partition the domain $[\mathbf{x}^L, \mathbf{x}^U]$ in $N > 1$ subdomains $\mathbf{B}_k$, calculate a convex under estimator $\hat{f}_k$ of f on each subdomain $\mathbf{B}_k$ with the basic α**BB** method, and then build up a global convex underestimator $\hat{f}$ on $\mathbf{B}$ from the partial $\hat{f}_k$'s.

For instance, in the *Spline* α**BB** method of Meyer and Floudas [13] one proceeds as follows: For each variable $x_i \in \mathbb{R}$, let the interval $[x_i^L, x_i^U]$ be partitioned into N_i subintervals with endpoints satisfying $x_i^L = x_i^0 < x_i^1 < \cdots < x_i^{N_i} = x_i^U$. A smooth convex underestimator ϕ over the domain $[\mathbf{x}^L, \mathbf{x}^U]$ is defined by:

$$\begin{aligned} \phi(\mathbf{x}) &:= f(\mathbf{x}) + q(\mathbf{x}) \\ \text{where} \quad q(\mathbf{x}) &:= \sum_{i=1}^{n} q_i^k(x_i) \text{ for } x_i \in \left[x_i^{k-1}, x_i^k\right], \\ q_i^k(x_i) &:= \alpha_i^k (x_i - x_i^{k-1})(x_i - x_i^k) + \beta_i^k x_i + \gamma_i^k. \end{aligned}$$

Continuity and smoothness of $q(\mathbf{x})$ are obtained as for spline methods, i.e., the functions q_i^k's and their derivatives must match at the endpoints x_i^k. In addition, one requires $q(\mathbf{x}_L) = q(\mathbf{x}_U) = 0$. So after the α_i^k's have been computed via the scaled Gershgorin method in each subdomain (x_i^k, x_i^{k+1}), this results in a linear system of equations on the β_i^k's and γ_i^k's of the form:

$$\begin{cases} \beta_i^1 x_i^0 + \gamma_i^1 = 0, \\ \beta_i^k x_i^k + \gamma_i^k = \beta_i^{k+1} x_i^k + \gamma_i^{k+1} \quad \forall k = 1, \ldots, N_i - 1, \\ \beta_i^{N_i} x_{N_i}^0 + \gamma_i^{N_i} = 0, \\ -\alpha_i^k (x_i^k - x_i^{k-1}) + \beta_i^k = -\alpha_i^{k+1} (x_i^{k+1} - x_i^k) + \beta_i^{k+1} \quad \forall k = 1, \ldots, N_i - 1. \end{cases}$$

The resulting convex underestimator ϕ on $\mathbf{B}$ is tighter than the one obtained in the α**BB** method directly applied on $\mathbf{B}$.

In the other refinement of α**BB** method by Gounaris and Floudas [7,8], one proceeds as follows: For chosen integers $N_i > 1, i = 1, 2, \ldots, n$, each interval $[x_i^L, x_i^U]$ is partitioned into N_i segments of equal length. The initial box $[\mathbf{x}^L, \mathbf{x}^U]$ is then partitioned into $N = \prod_{i=1}^n N_i$ subdomains of equal volume. And so the k^{th} subdomain is of the form $\mathbf{B}_k = [x_1^{k_1-1}, x_1^{k_1}] \times \cdots \times [x_n^{k_n-1}, x_n^{k_n}]$. For every subdomain $\mathbf{B}_k, k = 1, \ldots, N$, the corresponding α**BB** underestimator is of the form:

$$\mathcal{L}_k(\mathbf{x}) := f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i^k \left(x_i - x_i^{k_i-1}\right)\left(x_i - x_i^{k_i}\right),$$

where the value of each α_i^k is computed via the scaled Gershgorin method:

$$\alpha_i^k := \max\left\{0, -\frac{1}{2}\left(\underline{f}_{ii}^k - \sum_{j\neq i} \max\left\{|\underline{f}_{ji}^k|, |\overline{f}_{ji}^k|\right\}\right)\frac{\left(x_j^{k_j} - x_j^{k_j-1}\right)}{\left(x_i^{k_i} - x_i^{k_i-1}\right)}\right\},$$

in which $\{\underline{f}_{ji}^k, \overline{f}_{ji}^k\}$ are lower and upper bounds on $\partial^2 f/\partial x_j \partial x_i$, and $\underline{f}_{ii}^k$ is a lower bound on $\partial^2 f/\partial x_i^2$, in the subdomains $\mathbf{B}_k$; see Androulakis et al. [1].

Then a global convex underestimator on $\mathbf{B}$ is computed by combining these piecewise convex underestimators as detailed in [7,8].

As already mentioned, in both refinement methods [13] and [7,8] the quality of parameters α_i^k's in each of the subdomains is crucial for the overall performance of the resulting convex underestimator on $\mathbf{B}$. Therefore in view of the comparison in Sect. 4.3 between the standard α**BB** and the "moment" estimator that we propose, an alternative refinement of the standard α**BB** method is to proceed as in [13] or [7,8], but now in each subdomain $\mathbf{B}_k$ use a moment estimator as developed in Sect. 3 or in Sect. 4.4 instead of the standard α**BB** estimator.

Computational results. First, for illustration purpose, consider the Six-Hump Camelback function in Fig. 1 with the spline α**BB** underestimation method as computed in e.g. Gounaris and Floudas [8]. When the box $\mathbf{B} = [0, 1]^2$ is partitioned into 9 sub-boxes $\mathbf{B}_k$, $k = 1, \ldots, 9$, the resulting lower bound $f^*_{spline-\alpha\mathbf{BB}}$ is -1.28 which is strictly better than the lower bound -1.36 obtained with the (global) moment method. But if we use the moment method instead of the α**BB** method for each sub-box $\mathbf{B}_k$, then not only the resulting lower bound $f^*_{spline-mom} = -1.26$ is better, but also the new convex underestimator f_d is strictly "tighter" than the spline-α**BB** underestimator (as can be seen in Fig. 2). And indeed, $\int_{\mathbf{B}} |f - f_d| d\lambda = 0.28$ whereas $\int_{\mathbf{B}} |f - f_{spline-\alpha\mathbf{BB}}| d\lambda = 0.67$.

We have also compared the *spline*-α**BB** and the *spline-moment* underestimator when the original box $\mathbf{B}$ is partitioned in d^n sub-boxes of equal volume (i.e., with d segments of equal length on each coordinate axis) with up to $n = 6$ variables. As d^n grows very fast with n, in our examples d takes the values 2 and 3 only. Table 5 displays the results on some examples taken from Table 1 and the notation $f_1^* = f^*_{\alpha\mathbf{BB}}$, $f_2^* = f^*_{spline-\alpha\mathbf{BB}}$, $f_3^* = f^*_{spline-mom}$, $f_1^* = f^*_{mom}$, stand for the lower bounds obtained by minimizing on $\mathbf{B}$, the basic α**BB**, the spline-α**BB**, the spline-moment and the basic moment underestimators, respectively. The results confirm that the spline-α**BB** lower bounds $f^*_{spline-\alpha\mathbf{BB}}$ are significantly better than the basic α**BB** lower bounds f_1^*, and as expected, the lower bounds $f^*_{spline-mom}$ are strictly better than $f^*_{spline-\alpha\mathbf{BB}}$. In addition, in many cases the lower bounds f^*_{mom} are closer to the global minimum f^* than $f^*_{spline-\alpha\mathbf{BB}}$.

5 Parametric convex underestimators

We next build up convex underestimators f^s on the box $\mathbf{B}_s := [0, s]^n$ whose size is parameterized by the scalar $s \in [0, S]$ for some S. That is, instead of repeatedly computing a convex underestimator f^s for every s in some interval say $[0, S]$, one wishes to compute in one single shot (or once and for all $s \in [0, S]$), a parameterized family of best degree-d convex polynomial underestimators $(f_d^s) \subset \mathbb{R}[\mathbf{x}]_d$ of f on $\mathbf{B}_s$, $s \in [0, S]$.

Since the "tightness" criterion $\| f - f^s\|_1$ is a good indicator of the quality of the convex underestimator f^s on the box $[0, s]^n$, and in light of Sect. 3, a natural criterion to evaluate the efficiency of a parameterized convex underestimator f^s is the integral

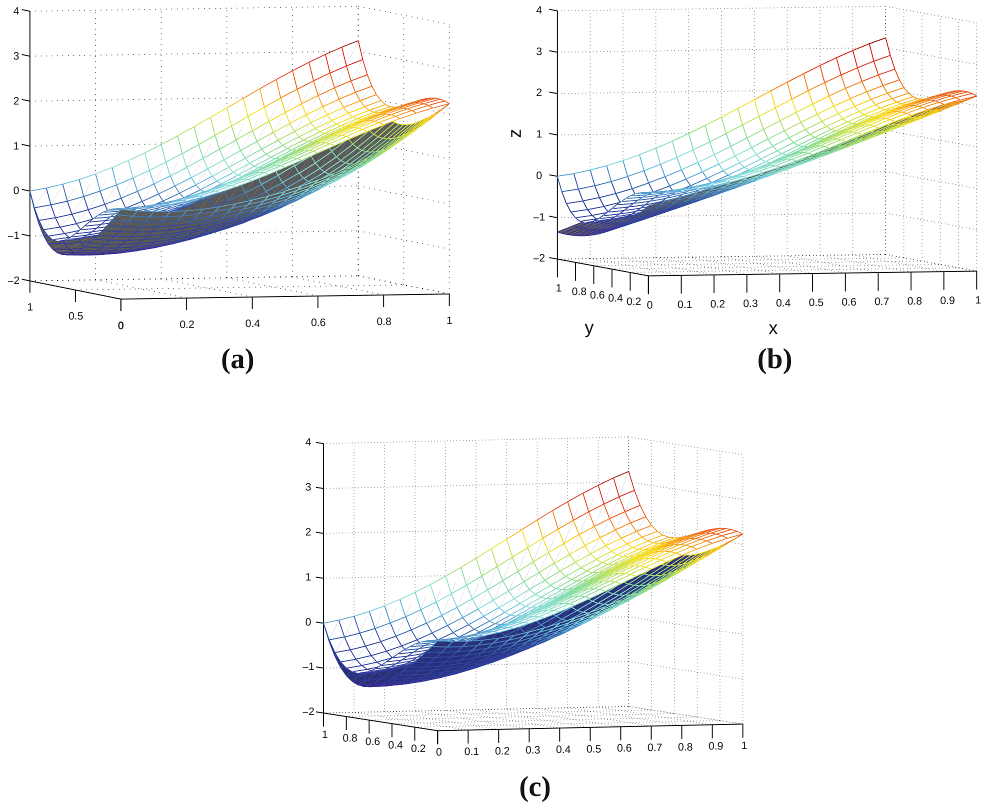
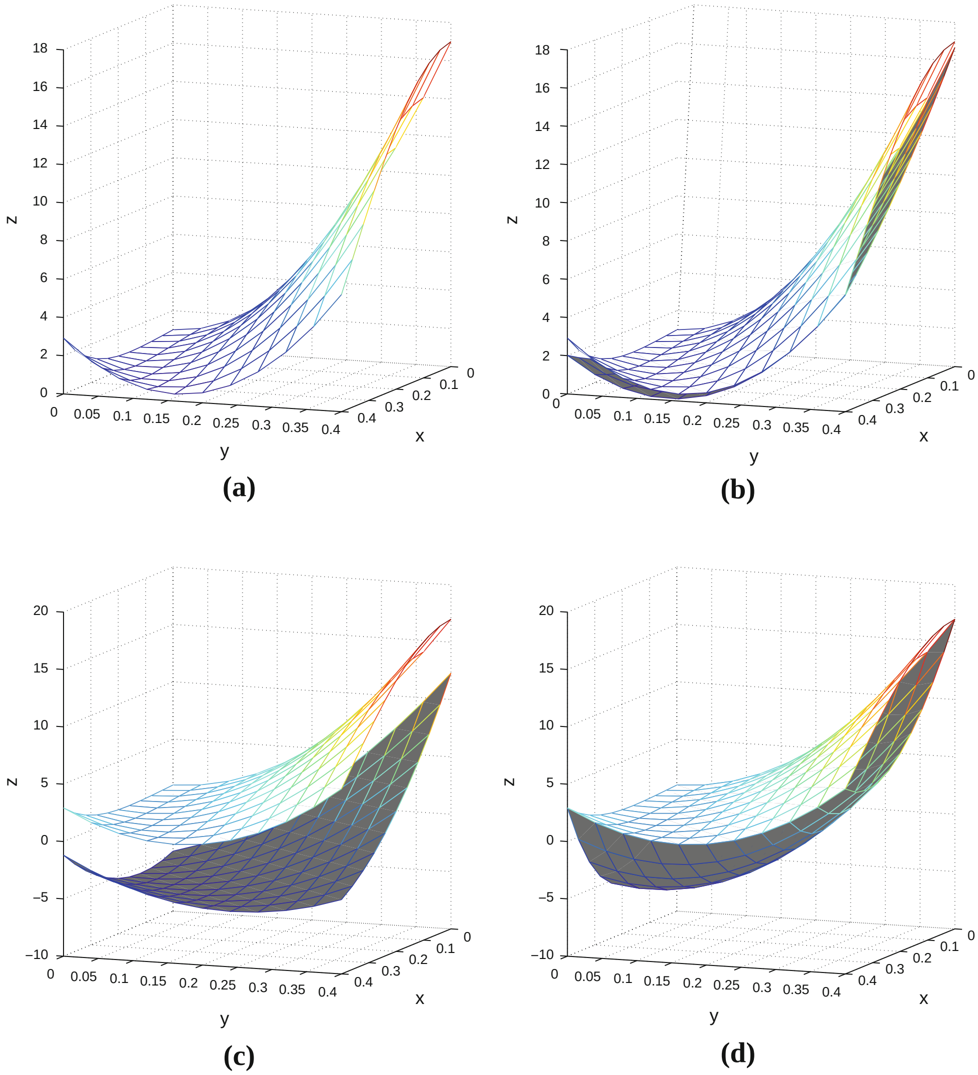


Fig. 2 Bivariate polynomial case. **a** $f^* = -1$; $f^*_{spline-\alpha\mathbf{BB}} = -1.28$, **b** $f^* = -1$; $f^*_{mom} = -1.36$, **c** $\mathbf{f}^*_{spline-mom} = -1.26$

Table 5 d: number of sub-intervals in each coordinate axis; $f_1^* = f^*_{\alpha\mathbf{BB}}$, $f_2^* = f^*_{spline-\alpha\mathbf{BB}}$, $f_3^* = f^*_{spline-mom}$, $f_4^* = f^*_{mom}$

Prob	n	deg f	d	$[\mathbf{x}^L, \mathbf{x}^U]$	f_1^*	f_2^*	f_3^*	f_4^*	f^*
Test2	4	3	3	[0,1]	−1.54	−1.006	−1.001	−1.22	−1
Test3	5	4	3	[−1,1]	−15	−6.07	−6	−13.95	−6
Test4	6	6	2	[−1,1]	−60.15	−18.5	−4.05	−10.06	−3
Test5	3	6	3	[−2,2]	−411.2	−37.79	−12.17	−12.66	−1
Test10	4	4	3	[0,1]	−197.54	−14.78	−0.8	−0.9698	0
Test11	4	4	3	[0,1]	−33.02	−19.77	−0.007	−0.623	0
Test14(1)	3	4	3	[−5,2]	−2,409	−855	−300	−300	−300
Test14(2)	4	4	3	[−5,2]	−3,212	−1,141.2	−400	−400	−400
Test14(3)	5	4	3	[−5,2]	−4,015	−1,426.5	−500	−500	−500
Fl.2.2	5	2	3	[0,1]	−18.9	−18.77	−17	−18.9	−17
Fl.2.3	6	2	2	[0,1]	−5,270.9	−2,176.4	−361	−361.50	−361



(a)

(b)

(c)

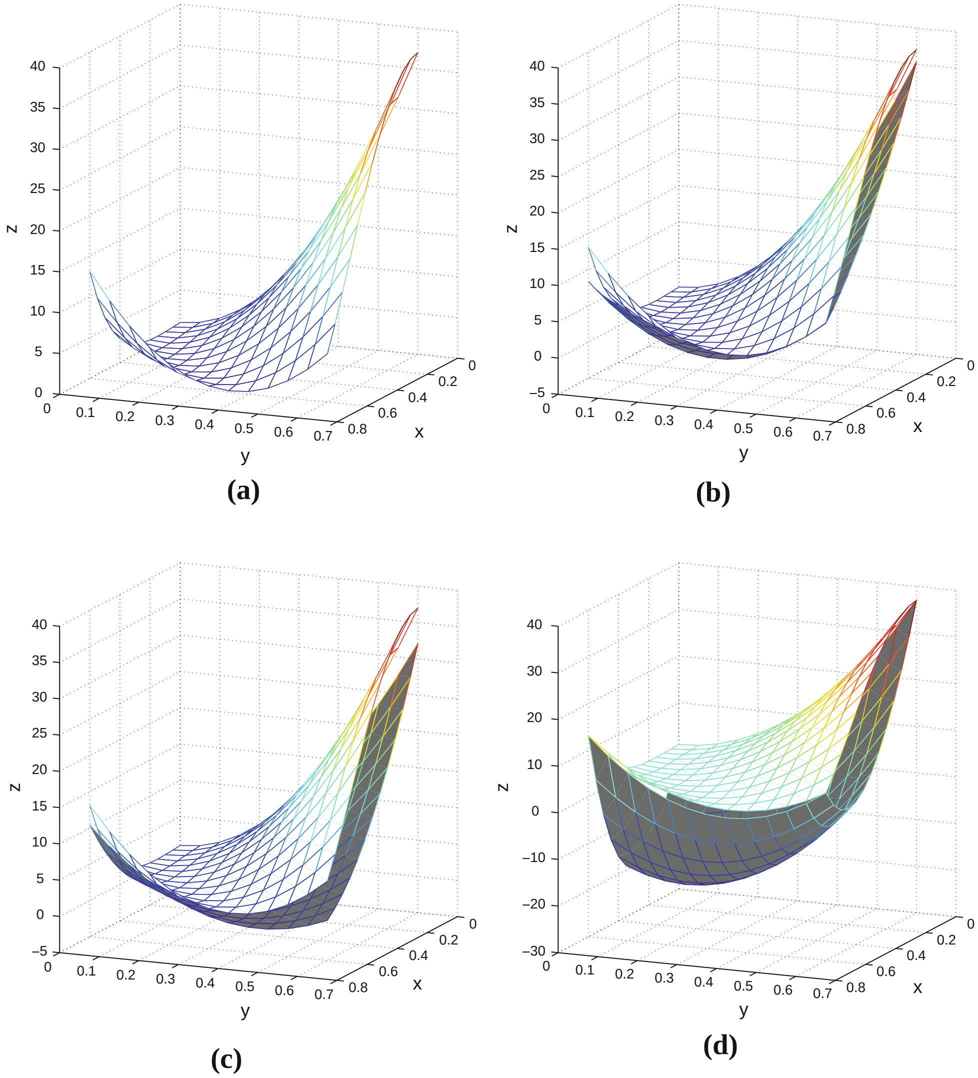
(d)

Fig. 3 Case $\mathbf{B}_s := [0, s]^2 = [0, 0.4]^2$. **a** $f^* = 0.36$, **b** $f^*_{mom} = 0.0883$, **c** $(f^{0.4}_{mom})^* = -5.5395$, **d** $f^*_{\alpha\mathbf{BB}} = -5.7240$

$$\int_0^S \| f - f^s \|_1 \, ds = \int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - f^s(\mathbf{x})) \, d\mathbf{x} \, ds.$$

For the dependence of f^s on the parameter s, a natural choice is to consider f^s as a degree-d polynomial in s and $\mathbf{x}$, i.e., $f^s_d \in \mathbb{R}[\mathbf{x}, s]_d$, optimal solution of the optimization problem:

$$\begin{cases} \rho_d = \min\limits_{h \in \mathbb{R}[\mathbf{x}, s]_d} \displaystyle\int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - h(\mathbf{x}, s)) \, d\mathbf{x} ds \\ \text{s.t.} \quad f - h \geq 0 \text{ on } \mathbf{B}_s \times [0, S] \\ \qquad\;\; \mathbf{T} h \geq 0 \text{ on } \mathbf{B}_s \times [0, S] \times \mathbf{U} \end{cases} \tag{16}$$



(a)

(b)

(c)

(d)

Fig. 4 Case $\mathbf{B}_s := [0, s]^2 = [0, 0.6]^2$. **a** $f^* = 0.1600$, **b** $f^*_{mom} = -0.5259$, **c** $(f^{0.6}_{mom})^* = -2.1770$, **d** $f^*_{\alpha\mathbf{BB}} = -22.117$

where the linear mapping $\mathbf{T} : \mathbb{R}[\mathbf{x}, s] \to \mathbb{R}[\mathbf{x}, s, \mathbf{y}]$ is now given by:

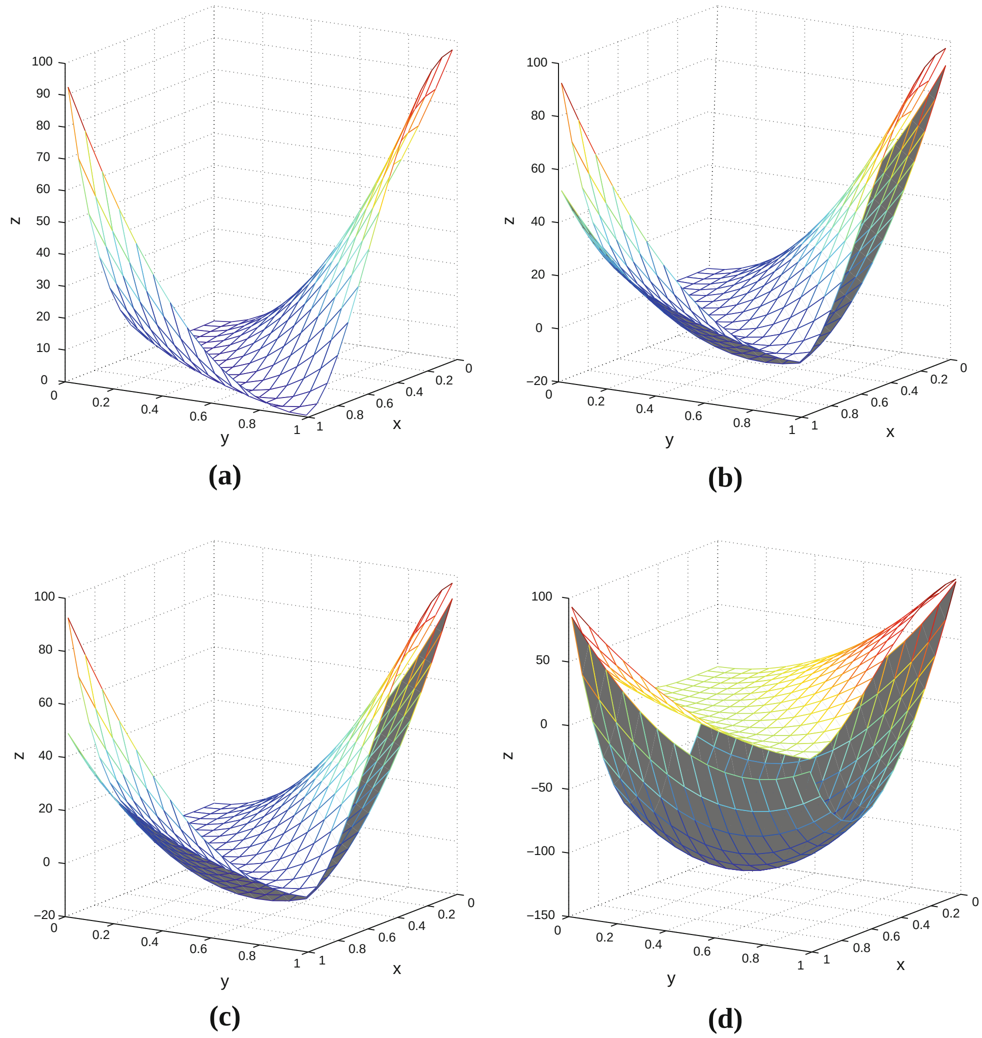
$$h \mapsto \mathbf{T}h(\mathbf{x}, s, \mathbf{y}) = \mathbf{y}' \, \nabla^2_{\mathbf{x}} h(\mathbf{x}, s) \, \mathbf{y}, \quad h \in \mathbb{R}[\mathbf{x}, s],$$

and $\nabla^2_{\mathbf{x}} h(\mathbf{x}, s) = (\partial^2 h(\mathbf{x}, s)/\partial x_i \partial x_j), i, j = 1, \ldots, n$.

Lemma 5.1 *For every* $d \geq \deg f$, *(16) has an optimal solution* $f^*_d \in \mathbb{R}[\mathbf{x}, s]$.

Proof Let $\Delta := \mathbf{B}_s \times [0, S]$. Notice that for every feasible solution $h \in \mathbb{R}[\mathbf{x}, s]_d$ of (16),

$$\int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - h(\mathbf{x}, s)) \, d\mathbf{x}ds = \int_{\Delta} |f(\mathbf{x}) - h(\mathbf{x}, s)| \, d\mathbf{x}ds = \|f - h\|_1,$$



(a) (b) (c) (d)

Fig. 5 Case $\mathbf{B}_s := [0, s]^2 = [0, 1]^2$. **a** $f^* = 0$, **b** $f^*_{mom} = -4.888$, **c** $(f^1_{mom})^* = -5.0467$, **d** $f^*_{\alpha\mathbf{BB}} = -121.76$

where $\|\cdot\|_1$ denote the L_1-norm on Δ. Therefore (16) reads

$$\begin{cases} \rho_d = \displaystyle\min_{h\in\mathbb{R}[\mathbf{x},s]_d} \|f-h\|_1 \\ \text{s.t.} \quad f-h \geq 0 \text{ on } \Delta\,; \quad \mathbf{T}h \geq 0 \text{ on } \Delta\times\mathbf{U}. \end{cases} \tag{17}$$

Hence with arguments similar to those used in the proof of Lemma 3.1, existence of a solution $f^*_d \in \mathbb{R}[\mathbf{x}, s]_d$ is guaranteed. □

For every $s \in [0, S]$, let $f^s_d \in \mathbb{R}[\mathbf{x}]$ be an optimal solution of (3) with $\mathbf{B}_s$ in lieu of $\mathbf{B}$, and consider the mapping $s \mapsto \rho_d(s) = \int_{\mathbf{B}_s} f^s_d(\mathbf{x})d\mathbf{x}$. With $f^*_d \in \mathbb{R}[\mathbf{x}, s]_d$ as in Lemma 5.1

Table 6 Comparing $f^*_{\alpha\mathbf{BB}}$, $(f^s_{mom})^*$, f^*_{mom} and f^* with $d = 2$

Prob	n	deg f	$[\mathbf{x}^L, \mathbf{x}^U]$	$f^*_{\alpha\mathbf{BB}}$	$(f^s_{mom})^*$	f^*_{mom}	f^*
Test14	3	4	[−5,1]	−1,726.0	−310.6	−300	−300
Test14	3	4	[−5,1.5]	−2,054.9	−335.5	−300	−300
Test14	3	4	[−5,2]	−2,408.9	−374.9	−300	−300
Test2	4	3	[0,0.2]	−0.21	−1.50	−0.2118	−0.2
Test2	4	3	[0,0.4]	−0.46	−1.03	−0.4423	−0.4
Test2	4	3	[0,0.6]	−0.73	−0.81	−0.6882	−0.6
Test2	4	3	[0,0.8]	−1.07	−0.94	−0.9496	−0.8
Test2	4	3	[0,1]	−1.55	−1.37	−1.2258	−1.0
Test10	4	4	[0,0.2]	50.74	16.17	50.75	50.75
Test10	4	4	[0,0.6]	3.09	18.76	26.29	28.83
Test10	4	4	[0,0.8]	−71.52	11.41	7.42	13.45
Test10	4	4	[0,1]	−197.55	−1.84	−0.96	0.00

$$\int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - f^*_d(\mathbf{x}, s))\, d\mathbf{x}\, ds \geq \rho^* := \int_0^S \int_{\mathbf{B}_s} (f(\mathbf{x}) - f^s_d(\mathbf{x}))\, d\mathbf{x}\, ds$$
$$= \int_\Delta f(\mathbf{x}) d\mathbf{x} ds - \int_0^S \rho_d(s)\, ds.$$

Ideally one would like to approximate ρ^*. This is possible if one considers polynomials $h(\mathbf{x}, s)$ with degree d in $\mathbf{x}$ *and* degree in s arbitrary large. Indeed, if $\rho_d(s)$ is Riemann integrable on $[0, S]$ then

$$\int_0^s \rho_d(s)\, ds = \lim_{N\to\infty} \frac{S}{N} \sum_{k=1}^N \rho_d(kS/N),$$

and one may find a polynomial $h_N \in \mathbb{R}[\mathbf{x}, s]$ of degree d in $\mathbf{x}$ and degree at most N in s, such that $h_N(\mathbf{x}, s) = f^s_d(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{B}_s$ and all $s = kS/N, k = 0, 1, \ldots, N$. Write $f^s_d(\mathbf{x}) = \sum_{\alpha\in\mathbb{N}^n_d} f^s_\alpha \mathbf{x}^\alpha$, for some coefficients (f^s_α), and for every $\alpha \in \mathbb{N}^n_d$ let $p_\alpha \in \mathbb{R}[s]_N$ be such that

$$p_\alpha(kS/N) = f^{kS/N}_\alpha, \quad k = 0, 1, \ldots, N.$$

Then the polynomial

$$(\mathbf{x}, s) \mapsto h_N(\mathbf{x}, s) := \sum_{\alpha\in\mathbb{N}^n_d} p_\alpha(s)\, \mathbf{x}^\alpha,$$

has degree N in s and degree d in $\mathbf{x}$, and matches f^s_d on $\mathbf{B}_{kS/N} \times \{kS/N\}, k = 0, \ldots, N$.

Semidefinite approximations. For practical purpose, $h \in \mathbb{R}[\mathbf{x}, s]_{2d}$ in (16) is a polynomial of degree d in $\mathbf{x}$ and each coefficient of $\mathbf{x}^\alpha$ is a polynomial of degree d in the parameter s, i.e.,

$$h(\mathbf{x}, s) = \sum_{\alpha\in\mathbb{N}^n_d} h_\alpha(s)\, \mathbf{x}^\alpha, \quad h_\alpha \in \mathbb{R}[s]_d. \tag{18}$$

And again, one approximates $f^*_{2d} \in \mathbb{R}[\mathbf{x}, s]_{2d}$ by a sequence of polynomials $(f_{dk}) \subset \mathbb{R}[\mathbf{x}, s]_{2d}, k \in \mathbb{N}$, of the form (18), by solving a hierarchy of semidefinite programs. Let $\hat{g}_j \in \mathbb{R}[\mathbf{x}, s]$, $j = 1, \ldots, n+2$ be the polynomials

$$s \mapsto \hat{g}_j(\mathbf{x}, s) := x_j(s - x_j), \quad j = 1, \ldots, n,$$

and

$$\hat{g}_{n+1}(\mathbf{x}, s) := s(S - s); \quad \hat{g}_{n+2}(\mathbf{x}, s, \mathbf{y}) := 1 - \|\mathbf{y}\|^2.$$

Consider the hierarchy of semidefinite programs:

$$\begin{aligned}
&\max_{h \in \mathbb{R}[\mathbf{x}, s]_{2d}, \sigma_j, \theta_k} \int_0^S \left(\int_{\mathbf{B}_s} h(\mathbf{x}, s)\, d\mathbf{x} \right) ds \\
\text{s.t.} \quad & f(\mathbf{x}) = h(\mathbf{x}, s) + \sum_{j=0}^{n+1} \sigma_j(\mathbf{x}, s)\hat{g}_j(\mathbf{x}, s) \quad \forall \mathbf{x} \\
& \mathbf{T}h(\mathbf{x}, s, \mathbf{y}) = \sum_{j=0}^{n+1} \theta_j(\mathbf{x}, s, \mathbf{y})\hat{g}_j(\mathbf{x}, s) \\
& \qquad + \theta_{n+2}(\mathbf{x}, s, \mathbf{y})\hat{g}_{n+2}(\mathbf{y}) \quad \forall \mathbf{x}, s, \mathbf{y} \\
& \sigma_0 \in \Sigma[\mathbf{x}, s]_k, \ \sigma_j \in \Sigma[\mathbf{x}, s]_{k-1}, \ j \geq 1 \\
& \theta_0 \in \Sigma[\mathbf{x}, s, \mathbf{y}]_k, \theta_j \in \Sigma[\mathbf{x}, s, \mathbf{y}]_{k-1}, \ j \geq 1,
\end{aligned} \tag{19}$$

with $k \geq \max[\lceil d/2 \rceil, \lceil (\deg f)/2 \rceil]$, and where $h \in \mathbb{R}[\mathbf{x}, s]_{2d}$ is of the form (18).

Of course, one may also consider more general boxes like e.g. two-parameter boxes of the form $\mathbf{B}_s := [s_1, s_2]^n$ with $0 \leq s_1 \leq s_2 \leq S$ for some fixed $S > 0$. But solving the semidefinite program (19) is computationally more expensive as the criterion now reads

$$\max_{h \in \mathbb{R}[\mathbf{x}, s_1, s_2]_{2d}, \sigma_j, \theta_k} \int_0^S \int_{s_1}^S \left(\int_{\mathbf{B}_s} h(\mathbf{x}, s_1, s_2)\, d\mathbf{x} \right) ds_2 ds_1,$$

and to define the constraints of (19) we also need introduce the polynomials

$$\hat{g}_j(\mathbf{x}, s_1, s_2) = (x_j - s_1)(s_2 - x_j), \quad j = 1, \ldots, n,$$

as well as $\hat{g}_{n+1}(\mathbf{x}, s_1, s_2) = s_1(S - s_1), \hat{g}_{n+2}(\mathbf{x}, s_1, s_2) := (s_2 - s_1)(S - s_2)$, and $\hat{g}_{n+3}(\mathbf{x}, s_1, s_2, \mathbf{y}) = 1 - \|\mathbf{y}\|^2$.

Computational results. We present experimental results in case where the convex polynomial underestimator is quadratic, i.e., $d = 2$ and k in (19) is set to its smallest possible value. All examples are taken from Gounaris and Floudas [8]. Given $f \in \mathbb{R}[\mathbf{x}]$ on the box $[0, 1]^n$, we try to find the best convex polynomial underestimator $h(\mathbf{x}, s)$ of degree $d = 2$ in the variable $\mathbf{x}$ and the coefficients of $\mathbf{x}$ are univariate polynomials of degree 2 in the parameter s; see (18).

The lower bounds $(f^s_{mom})^*$ of f on $\mathbf{B}_s$ obtained by minimizing the parametric underestimator f^s_{mom} on $\mathbf{B}_s$, are computed for some values of $s \in [0, 1]$. Figures 3, 4 and 5 display results for the simple bivariate example

$$\mathbf{x} \mapsto f(\mathbf{x}) = 1 - 2x_1 + x_1^2 + 100\, x_2^2 - 200\, x_1^2\, x_2 + 100\, x_1^4$$

in the box $\mathbf{B} = [0,1]^2$; three see [8]. The moment convex underestimator $f_d(\mathbf{x})$ of Sect. 3 is constructed for the three boxes $[0,0.4]^2$, $[0,0.6]^2$ and $[0,1]^2$. The parametric convex underestimator f^s_{mom} on the box $\mathbf{B}_s := [0,s]^2$ is computed for all $s \in [0,S]$ with $S = 5$, via solving (19).

Table 6 illustrates some examples with 4 variables from Gounaris and Floudas [8]. By increasing the size of box $\mathbf{B}_s := [0,s]^n$ with $s \in [0,5]$ or $s \in [-5,5]$, we compute the respective minima $(f^s_{mom})^*$, f^*_{mom} and f^* of the parametric convex underestimator f^s, the convex underestimator f_{mom} and the initial polynomial f.

Final remark. Notice that such convex underestimators f^s, $s \in [0,S]$, can be computed off-line before starting the search in the branch and bound search tree. But then these convex underestimators f^s are restricted to boxes $[0,s]^n$, $s \in [0,S]$. On the other hand, if the computation is done on-line at each node of the search tree, then one may compute parametric convex polynomial underestimators f^s on any box $[-s,s]^n(\text{or}[0,s]^n)$, $s \in [0,1]$, after the initial box $\mathbf{B} = \prod_{i=1}^n [a_i,b_i]$ is rescaled to $[-1,1]^n(\text{or}[0,1]^n)$ via a suitable change of variable.

6 Conclusion

We have provided an algorithm to compute convex polynomial underestimators of a given polynomial f on a box $\mathbf{B} \subset \mathbb{R}^n$. By solving a hierarchy of semidefinite programs one may approximate, as closely as desired, the best degree-d convex underestimator of a nonconvex polynomial f, for the L_1-norm. i.e., the one which minimizes the discrepancy $\int_{\mathbf{B}} |f - g| d\lambda$ over $\mathbf{B}$. On a sample of non convex problems from the literature, the lower bounds obtained (even at the first semidefinite program in the hierarchy), by minimizing this convex underestimator are significantly better than those obtained by minimizing the popular α**BB** underestimator. Finally we have also provided parametric convex polynomial underestimators $h \in \mathbb{R}[\mathbf{x}, s]$ where the parameter $s \in [0,S]$ defines the size of the box $[0,s]^n$ where one wishes to underestimate f by a convex polynomial $f^s \in \mathbb{R}[\mathbf{x}]$, namely $\mathbf{x} \mapsto f^s(\mathbf{x}) = h(\mathbf{x}, s)$ with s fixed. And so in mixed integer non linear programs, this convex underestimator can be computed off-line, which permits to avoid computing on-line at each node of the search tree of the Branch and Bound, a convex underestimator on $[0,s]^n$, for each desired value of $s \in [0,S]$.

Acknowledgments The authors wish to thank anonymous referees for very helpful comments and suggestions to improve the initial version of this paper.

Appendix

Recall that the *moment* matrix $\mathbf{M}_k(\mathbf{y})$ associated with a sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n_{2k}$, is the real symmetric matrix with rows and columns indexed by the monomial basis $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n_k$, and with entries

$$\mathbf{M}_k(\mathbf{y})[\alpha, \beta] = y_{\alpha+\beta}, \quad \forall \alpha, \beta \in \mathbb{N}^n_k.$$

Similarly, the *localizing* matrix $\mathbf{M}_k(\mathbf{y}\, g_j)$ associated with the polynomial $\mathbf{x} \mapsto g(\mathbf{x}) := \sum_\gamma g_\gamma \mathbf{x}^\gamma$ and with a sequence $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n_{2k}$, is the real symmetric matrix with rows and columns indexed by the monomial basis $(\mathbf{x}^\alpha)$, $\alpha \in \mathbb{N}^n_k$, and with entries

$$\mathbf{M}_k(g\,\mathbf{y})[\alpha,\beta] \;=\; \sum_{\gamma\in\mathbb{N}^n} g_\gamma\, y_{\alpha+\beta+\gamma}, \quad \forall\,\alpha,\beta\in\mathbb{N}^n_k.$$

For more details on moment and localizing matrices the interested reader is referred to e.g. Lasserre [12].

Proof of Lemma 3.2

Proof From the definition (1) of $\mathbf{T}$, define the operator $\mathbf{T}^* : \mathbb{R}[\mathbf{x},\mathbf{y}]^*_{2k} \to \mathbb{R}[\mathbf{x}]^*_{2k}$ by:

$$\langle g, \mathbf{T}^*\,\mathbf{z}\rangle \;=\; \langle \mathbf{T}\,g, \mathbf{z}\rangle, \quad \forall g\in\mathbb{R}[\mathbf{x}]_{2k},\; \mathbf{z}\in\mathbb{R}[\mathbf{x},\mathbf{y}]^*_{2k}.$$

The dual of (6) is a semidefinite program whose feasible set is described by:

$$\begin{array}{l} \mathbf{M}_k(\mathbf{u}),\; \mathbf{M}_{k-1}(g_j\,\mathbf{u}) \succeq 0, \quad j=1,\ldots,n\\ \mathbf{M}_k(\mathbf{z}),\; \mathbf{M}_{k-1}(g_j\,\mathbf{z}) \succeq 0, \quad j=1,\ldots,n+1\\ -(\mathbf{T}^*\mathbf{z})_\alpha + u_\alpha = \gamma_\alpha, \quad \forall\alpha\in\mathbb{N}^n_d, \end{array} \tag{20}$$

where:

- $\mathbf{u}\in\mathbb{R}[\mathbf{x}]^*_{2k}$, $\mathbf{z}\in\mathbb{R}[\mathbf{x},\mathbf{y}]^*_{2k}$ and $\gamma_\alpha = \int_{\mathbf{B}} \mathbf{x}^\alpha d\mathbf{x}$ for every $\alpha\in\mathbb{N}^n$, and
- $\mathbf{M}_k(\mathbf{u})$ (resp. $\mathbf{M}_{k-1}(g_j\,\mathbf{u})$) is the moment (resp. localizing) matrix associated with the sequence $\mathbf{u}$ and the polynomial g_j. And similarly for $\mathbf{M}_k(\mathbf{z})$ and $\mathbf{M}_{k-1}(g_j\,\mathbf{z})$.

It suffices to show that the set (20) has a strictly feasible solution $(\mathbf{u},\mathbf{z})$, in which case (a) there is no duality gap between (6) and its dual, and (b) the dual has an optimal solution if the optimal value is finite. So with $\epsilon>0$ fixed, let $\mathbf{z}\in\mathbb{R}[\mathbf{x},\mathbf{y}]^*_{2k}$ be the moment sequence associated with the Borel measure $\epsilon\cdot\lambda\otimes\nu$ on $\mathbf{B}\times\mathbf{U}$, where λ and ν are Borel probability measures uniformly supported on $\mathbf{B}$ and $\mathbf{U}$ respectively. Hence $\mathbf{M}_k(\mathbf{z}), \mathbf{M}_{k-1}(g_j\,\mathbf{z})\succ 0$, $j=1,\ldots,n+1$, because both $\mathbf{U}$ and $\mathbf{B}$ have nonempty interior. Observe that

$$-(\mathbf{T}^*\mathbf{z})_\alpha \;=\; \langle -\mathbf{x}^\alpha, \mathbf{T}^*\,\mathbf{z}\rangle \;=\; -\epsilon\,\theta_\alpha,$$

with

$$\theta_\alpha \;:=\; \sum_{i,j=1}^n \int_{\mathbf{B}} \frac{\partial^2\mathbf{x}^\alpha}{\partial x_i\,\partial x_j} d\lambda(\mathbf{x}) \int_{\mathbf{U}} y_i y_j\, d\nu(\mathbf{y}), \quad \forall\,\alpha\in\mathbb{N}^n_{2k}.$$

In particular, $(\mathbf{T}^*\mathbf{z})_\alpha = 0$ whenever $|\alpha|<2$. Next, let $\mathbf{u}\in\mathbb{R}[\mathbf{x}]^*_{2k}$ be such that

$$u_\alpha \;=\; \begin{cases} \gamma_\alpha + \epsilon\,\theta_\alpha & \forall\,\alpha\in\mathbb{N}^n_d\\ \gamma_\alpha & \forall\,\alpha\in\mathbb{N}^n_{2k},\; |\alpha|>d. \end{cases} \tag{21}$$

So the linear constraints $-(\mathbf{T}^*\mathbf{z})_\alpha + u_\alpha = \gamma_\alpha$ of (20) are all satisfied. Moreover, from (21), the moment matrix $\mathbf{M}_k(\mathbf{u})$ reads $\mathbf{M}_k(\gamma)+\epsilon\Delta_k$ for some matrix Δ_k, and similarly, the localizing matrix $\mathbf{M}_{k-1}(g_j\,\mathbf{u})$ reads $\mathbf{M}_{k-1}(g_j\,\mathbf{u}) = \mathbf{M}_{k-1}(g_j\,\gamma)+\epsilon\Theta_{jk}$ for some appropriate matrix Θ_{jk}, $j=1,\ldots,n$. Since $\mathbf{M}_k(\gamma)\succ 0$ and $\mathbf{M}_{k-1}(g_j\,\gamma)\succ 0$ we also have $\mathbf{M}_k(\mathbf{u})\succ 0$ and $\mathbf{M}_{k-1}(g_j\,\mathbf{u})\succ 0$ provided that $\epsilon>0$ is sufficiently small. Hence we have found a strictly feasible solution $(\mathbf{u},\mathbf{z})$ for the set (20).

We next prove that (6) has a feasible solution. With $\delta>0$ and k sufficiently large, the constant function $h := f^*-\delta\in\mathbb{R}[\mathbf{x}]_d$ is feasible for (6). Indeed, by Putinar's Positivstellensatz, $f-f^*+\delta$ being strictly positive on $\mathbf{B}$, it can be written $f-f^*+\delta = \sum_{j=0}^n \sigma_j g_j$ for some SOS polynomials $(\sigma_j)\in\Sigma_{k-1}[\mathbf{x}]$. And since $\mathbf{T}h=0$ one also has $\mathbf{T}h = \sum_{j=0}^{n+1}\theta_j g_j$ with $\theta_j = 0$ for all j. Finally, every feasible solution h of (6) satisfies $h\le f$ on $\mathbf{B}$, and so the objective value $\int_{\mathbf{B}} h d\mathbf{x}$ is bounded above by $\int_{\mathbf{B}} f d\mathbf{x}$ which is finite. Hence (6) has an optimal solution. □

References

1. Androulakis, I.P., Maranas, C.D., Floudas, C.A.: α**BB**: a global optimization method for general constrained nonconvex problems. J. Glob. Optim. **7**, 337–363 (1995)
2. Akrotirianakis, I.G., Floudas, C.A.: A new class of improved convex underestimators for twice continuously differentiable constrained NLPs. J. Glob. Optim. **30**, 367–390 (2004)
3. Cafieri, S., Lee, J., Liberti, L.: On convex relaxations of quadrilinear terms. J. Glob. Optim. **47**, 661–685 (2010)
4. Floudas, C.A., Pardalos, P.M., Adjiman, C., Esposito, W.R., Gümüş, Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A., Schweiger, C.A.: Handbook of Test Problems in Local and Global Optimization. Kluwer, Dordrecht (1999)
5. Floudas, C.A.: Deterministic Global Optimization Theory, Methods and Applications. Kluwer, Dordrecht (2000)
6. Floudas, C.A., Pardalos, P. (eds.): Encyclopedia of Optimization. Kluwer, Dordrecht (2001)
7. Gounaris, C.E., Floudas, C.A.: Tight convex underestimators for C^2-continuous problems:I. Univariate functions. J. Glob. Optim. **42**, 51–67 (2008)
8. Gounaris, C.E., Floudas, C.A.: Tight convex underestimators for C^2-continuous problems:II. Multivariate functions. J. Glob. Optim. **42**, 69–89 (2008)
9. Henrion, D., Lasserre, J.B., Lofberg, J.: GloptiPoly 3: moments, optimization and semidefinite programming. Optim. Methods Softw. **24**, 761–779 (2009)
10. Kramer, W., Geulig, I.: Interval Calculus in Maple. Wissenschaftliches Rechnen Bergische Universitat, GH Wuppertal (2001)
11. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**, 796–817 (2001)
12. Lasserre, J.B.: Moments, Positive Polynomials and Their Applications. Imperial College Press, UK (2009)
13. Meyer, C.A., Floudas, C.A.: Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: spline aBB underestimators. J. Glob. Optim. **32**, 221–258 (2005)
14. Putinar, M.: Positive polynomials on compact semi-algebraic sets. Indiana Univ. Math. J. **42**, 969–984 (1993)
15. Tawarmalani, M., Sahinidis, N.V.: Convex extensions and envelopes of lower semi-continuous functions. Math. Program. **93**, 247–263 (2002)
16. Waki, H., Kim, S., Kojima, M., Maramatsu, M.: Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. SIAM J. Optim. **17**, 218–242 (2006)
17. Wade, W.R.: The bounded convergence theorem. Am. Math. Mon. **81**, 387–389 (1974)

Appendix B

Paper: “A “Joint+marginal" algorithm for polynomial optimization"

A “joint+marginal” algorithm for polynomial optimization

Jean B. Lasserre and Tung Phan Thanh

***Abstract*— We present a new algorithm for solving a polynomial program $\mathbf{P}$ based on the recent “joint + marginal” approach of the first author for parametric polynomial optimization. The idea is to first consider the variable x_1 as a *parameter* and solve the associated $(n-1)$-variable $(x_2,\ldots,x_n)$ problem $\mathbf{P}(x_1)$ where the parameter x_1 is fixed and takes values in some interval $\mathbf{Y}_1 \subset \mathbb{R}$, with some probability φ_1 uniformly distributed on $\mathbf{Y}_1$. Then one considers the hierarchy of what we call “joint+marginal” semidefinite relaxations, whose duals provide a sequence of univariate polynomial approximations $x_1 \mapsto p_k(x_1)$ that converges to the optimal value function $x_1 \mapsto J(x_1)$ of problem $\mathbf{P}(x_1)$, as k increases. Then with k fixed *à priori*, one computes $\tilde{x}_1^* \in \mathbf{Y}_1$ which minimizes the univariate polynomial $p_k(x_1)$ on the interval $\mathbf{Y}_1$, a convex optimization problem that can be solved via a single semidefinite program. The quality of the approximation depends on how large k can be chosen (in general for significant size problems $k = 1$ is the only choice). One iterates the procedure with now an $(n-2)$-variable problem $\mathbf{P}(x_2)$ with parameter x_2 in some new interval $\mathbf{Y}_2 \subset \mathbb{R}$, etc. so as to finally obtain a vector $\tilde{\mathbf{x}} \in \mathbb{R}^n$. Preliminary numerical results are provided.**

I. Introduction

Consider the general polynomial program

$$\mathbf{P}: \quad f^* := \min_{\mathbf{x}} \{ f(\mathbf{x}) \;:\; \mathbf{x} \in \mathbf{K} \} \tag{1}$$

where f is a polynomial, $\mathbf{K} \subset \mathbb{R}^n$ is a basic semi-algebraic set, and f^* is the *global* minimum of $\mathbf{P}$ (as opposed to a local minimum). One way to approximate the global optimum f^* of $\mathbf{P}$ is to solve a hierarchy of either LP-relaxations or semidefinite relaxations as proposed in e.g. Lasserre [4], [5]. Despite computational experiments seem to reveals that convergence is fast, the matrix size in the i-th semidefinite relaxation of the hierarchy grows up as fast as $O(n^i)$. Hence, for large size (and sometimes even medium size) problems, only a few relaxations of the hierarchy can be implemented (the first, second or third relaxation). In that case, one only obtains a lower bound on f^*, and no feasible solution in general. So an important issue is:

How can we use the result of the i-th semidefinite relaxation to find an approximate feasible solution of the original problem?

For some well-known special cases of 0/1 optimization like e.g. the celebrated MAXCUT problem, one may generate a feasible solution with guaranteed performance, from a randomized rounding procedure that uses an optimal solution of the first semidefinite relaxation (i.e. with $i = 1$); see Goemans and Williamson [2]. But in general there is no such procedure.

Our contribution is to provide two relatively simple algorithms for polynomial programs which build up upon the so-called “joint+marginal” approach (in short (J+M)) developed in [6] for *parametric* polynomial optimization. The (J+M)-approach for variables $\mathbf{x} \in \mathbb{R}^n$ and parameters $\mathbf{y}$ in a simple set $\mathbf{Y}$, consists of the standard hierarchy of semidefinite relaxations in [4] where one treats the parameters $\mathbf{y}$ also as variables. But now the moment-approach implemented in the semidefinite relaxations, considers a *joint* probability distribution on the pair $(\mathbf{x}, \mathbf{y})$, with the additional constraint that the *marginal* distribution on $\mathbf{Y}$ is fixed (e.g. the uniform probability distribution on $\mathbf{Y}$); whence the name *“joint+marginal”*.

For every $k = 1,\ldots,n$, let the compact interval $\mathbf{Y}_k := [\underline{x}_k, \overline{x}_k] \subset \mathbb{R}$ be contained in the projection of $\mathbf{K}$ into the x_k-coordinate axis. In the context of the (non-parametric) polynomial optimization (1), the above (J+M)-approach can be used as follows in what we call the **(J+M)-algorithm**:

- (a) Treat x_1 as a parameter in the compact interval $\mathbf{Y}_1 = [\underline{x}_1, \overline{x}_1]$ with associated probability distribution φ_1 uniformly distributed on $\mathbf{Y}_1$.
- (b) **with** $i \in \mathbb{N}$ **fixed**, solve the i-th semidefinite relaxation of the (J+M)-hierarchy [6] applied to problem $\mathbf{P}(x_1)$ with $n-1$ variables $(x_2,\ldots,x_n)$ and parameter x_1, which is problem $\mathbf{P}$ with the additional constraint that the variable $x_1 \in \mathbf{Y}_1$ is fixed. The dual provides a univariate polynomial $x_1 \mapsto J_i^1(x_1)$ which, if i would increase, would converge to $J^1(x_1)$ in the $L_1(\varphi_1)$-norm. (The map $v \mapsto J^1(v)$ denotes the optimal value function of $\mathbf{P}(v)$, i.e. the optimal value of $\mathbf{P}$ given that the variable x_1 is fixed at the value v.) Next, compute $\tilde{x}_1 \in \mathbf{Y}_1$, a global minimizer of the univariate polynomial J_i^1 on $\mathbf{Y}_1$ (e.g. this can be done by solving a single semidefinite program). Ideally, when i is large enough, $\tilde{x}_1$ should be close to the first coordinate x_1^* of a global minimizer $\mathbf{x}^* = (x_1^*,\ldots,x_n^*)$ of $\mathbf{P}$.
- (c) go back to step (b) with now $x_2 \in \mathbf{Y}_2 \subset \mathbb{R}$ instead of x_1, and with φ_2 being the probability measure uniformly distributed on $\mathbf{Y}_2$. With the same method, compute a global minimizer $\tilde{x}_2 \in \mathbf{Y}_2$, of the univariate polynomial $x_2 \mapsto J_i^2(x_2)$ on the interval $\mathbf{Y}_2$. Again, if i would increase, J_i^2 would converge in the $L_1(\varphi_2)$-norm to the optimal value function $v \mapsto J^2(v)$ of $\mathbf{P}(x_2)$ (i.e. the optimal value of $\mathbf{P}$ given that the variable x_2 is fixed at the value v.) Iterate until one has obtained $\tilde{x}_n \in \mathbf{Y}_n \subset \mathbb{R}$.

One ends up wih a point $\tilde{\mathbf{x}} \in \prod_{k=1}^n \mathbf{Y}_k$ and in general $\tilde{\mathbf{x}} \notin \mathbf{K}$. One may then use $\tilde{\mathbf{x}}$ as initial guess of a local optimization procedure to find a local minimum $\hat{\mathbf{x}} \in \mathbf{K}$.

J.B. Lasserre is with LAAS-CNRS and the Institute of Mathematics, University of Toulouse, France. lasserre@laas.fr

Tung Phan Thanh is with LAAS-CNRS, University of Toulouse, France. tphanth@laas.fr

The rationale behind the (J+M)-algorithm is that if i is large enough and $\mathbf{P}$ has a unique global minimizer $\mathbf{x}^* \in \mathbf{K}$, then $\tilde{\mathbf{x}}$ as well as $\hat{\mathbf{x}}$ should be close to $\mathbf{x}^*$.

The computational complexity before the local optimization procedure is less than solving n times the i-th semidefinite relaxation in the (J+M)-hierarchy (which is itself of same order as the i-th semidefinite relaxation in the hierarchy defined in [4]), i.e., a polynomial in the input size of $\mathbf{P}$.

When the feasible set $\mathbf{K}$ is convex, one may define the following variant to obtain a *feasible* point $\tilde{\mathbf{x}} \in \mathbf{K}$. Again, let $\mathbf{Y}_1$ be the projection of $\mathbf{K}_1$ into the x_1-coordinate axis. Once $\tilde{x}_1 \in \mathbf{Y}_1$ is obtained in step (b), consider the new optimization problem $\mathbf{P}(\tilde{x}_1)$ in the $n-1$ variables $(x_2, \ldots, x_n)$, obtained from $\mathbf{P}$ by fixing the variable $x_1 \in \mathbf{Y}_1$ at the value $\tilde{x}_1$. Its feasible set is the convex set $\mathbf{K}_1 := \mathbf{K} \cap \{\mathbf{x} : x_1 = \tilde{x}_1\}$. Let $\mathbf{Y}_2$ be the projection of $\mathbf{K}_1$ into the x_2-coordinate axis. Then go back to step (b) with now $x_2 \in \mathbf{Y}_2$ as parameter and $(x_3, \ldots, x_n)$ as variables, to obtain a point $\tilde{x}_2 \in \mathbf{Y}_2$, etc. until a point $\tilde{\mathbf{x}} \in \prod_{k=1}^n \mathbf{Y}_k$ is obtained. Notice that now $\tilde{\mathbf{x}} \in \mathbf{K}$ because $\mathbf{K}$ is convex. Then proceed as before with $\tilde{\mathbf{x}}$ being the initial guess of a local minimization algorithm to obtain a local minimizer $\hat{\mathbf{x}} \in \mathbf{K}$ of $\mathbf{P}$.

II. The "joint+marginal" approach to parametric optimization

Most of the material of this section is taken from [6]. Let $\mathbb{R}[\mathbf{x}, \mathbf{y}]$ denote the ring of polynomials in the variables $\mathbf{x} = (x_1, \ldots, x_n)$, and the variables $\mathbf{y} = (y_1, \ldots, y_p)$, whereas $\mathbb{R}[\mathbf{x}, \mathbf{y}]_d$ denotes its subspace of polynomials of degree at most d. Let $\Sigma[\mathbf{x}, \mathbf{y}] \subset \mathbb{R}[\mathbf{x}, \mathbf{y}]$ denote the subset of polynomials that are sums of squares (in short s.o.s.). For a real symmetric matrix $\mathbf{A}$ the notation $\mathbf{A} \succeq 0$ stands for $\mathbf{A}$ is positive semidefinite.

A. The parametric optimization problem

Let $\mathbf{Y} \subset \mathbb{R}^p$ be a compact set, called the *parameter* set, and let $f, h_j \in \mathbb{R}[\mathbf{x}]$, $j = 1, \ldots, m$. Let $\mathbf{\Delta} \subset \mathbb{R}^n \times \mathbb{R}^p$ be the basic closed semi-algebraic set:

$$\mathbf{\Delta} := \{(\mathbf{x}, \mathbf{y}) \,:\, \mathbf{y} \in \mathbf{Y}\,;\, h_j(\mathbf{x}, \mathbf{y}) \geq 0,\, j = 1, \ldots, m\} \quad (2)$$

and for each $\mathbf{y} \in \mathbf{Y}$, let

$$\mathbf{\Delta}_{\mathbf{y}} \,:=\, \{\, \mathbf{x} \in \mathbb{R}^n \,:\, (\mathbf{x}, \mathbf{y}) \,\in\, \mathbf{\Delta} \,\}. \quad (3)$$

For each $\mathbf{y} \in \mathbf{Y}$, fixed, consider the optimization problem:

$$J(\mathbf{y}) \,:=\, \inf_{\mathbf{x}} \{\, f(\mathbf{x}, \mathbf{y}) \,:\, (\mathbf{x}, \mathbf{y}) \,\in\, \mathbf{\Delta} \,\}. \quad (4)$$

The interpretation is as follows: $\mathbf{Y}$ is a set of parameters and for each instance $\mathbf{y} \in \mathbf{Y}$ of the parameter, one wishes to compute an optimal *decision* vector $\mathbf{x}^*(\mathbf{y})$ that solves problem (4). Let φ be a Borel probability measure on $\mathbf{Y}$, with a positive density with respect to the Lebesgue measure on $\mathbb{R}^p$ (or with respect to the counting measure if $\mathbf{Y}$ is discrete). For instance

$$\varphi(B) \,:=\, \left(\int_{\mathbf{Y}} d\mathbf{y} \right)^{-1} \int_{\mathbf{Y} \cap B} d\mathbf{y}, \qquad \forall B \in \mathcal{B}(\mathbb{R}^p),$$

is uniformly distributed on $\mathbf{Y}$. Sometimes, e.g. in the context of optimization with data uncertainty, φ is already specified. The idea is to use φ (or more precisely, its moments) to get information on the distribution of optimal solutions $\mathbf{x}^*(\mathbf{y})$ of $\mathbf{P}_{\mathbf{y}}$, viewed as random vectors. In this section we assume that for every $\mathbf{y} \in \mathbf{Y}$, the set $\mathbf{\Delta}_{\mathbf{y}}$ in (3) is nonempty.

B. A related infinite-dimensional linear program

Let $\mathbf{M}(\mathbf{\Delta})$ be the set of finite Borel probability measures on $\mathbf{\Delta}$, and consider the following infinite-dimensional linear program $\mathbf{P}$:

$$\rho \,:=\, \inf_{\mu \in \mathbf{M}(\mathbf{\Delta})} \left\{ \int_{\mathbf{\Delta}} f \, d\mu \,:\, \pi\mu \,=\, \varphi \right\}, \quad (5)$$

where $\pi\mu$ denotes the marginal of μ on $\mathbb{R}^p$, that is, $\pi\mu$ is a probability measure on $\mathbb{R}^p$ defined by $\pi\mu(B) := \mu(\mathbb{R}^n \times B)$ for all $B \in \mathcal{B}(\mathbb{R}^p)$. The dual of $\mathbf{P}$ is the following infinite-dimensional linear program:

$$\begin{array}{rll} \rho^* \,:=\, \displaystyle\sup_{p \in \mathbb{R}[\mathbf{y}]} & \displaystyle\int_{\mathbf{Y}} p(\mathbf{y}) \, d\varphi(\mathbf{y}) & \\ & f(\mathbf{x}) - p(\mathbf{y}) \,\geq\, 0 \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathbf{\Delta}. \end{array} \quad (6)$$

Theorem 1 ([6]): Let both $\mathbf{Y} \subset \mathbb{R}^p$ and $\mathbf{\Delta}$ in (2) be compact and assume that for every $\mathbf{y} \in \mathbf{Y}$, the set $\mathbf{\Delta}_{\mathbf{y}} \subset \mathbb{R}^n$ in (3) is nonempty. Let $\mathbf{P}$ be the optimization problem (5) and let $\mathbf{X}^*_{\mathbf{y}} := \{\mathbf{x} \in \mathbb{R}^n \,:\, f(\mathbf{x}, \mathbf{y}) = J(\mathbf{y})\}$, $\mathbf{y} \in \mathbf{Y}$. Then:

(a) $\rho = \displaystyle\int_{\mathbf{Y}} J(\mathbf{y}) \, d\varphi(\mathbf{y})$ and $\mathbf{P}$ has an optimal solution.

(b) Assume that for φ-almost $\mathbf{y} \in \mathbf{Y}$, the set of minimizers of $\mathbf{X}^*_{\mathbf{y}}$ is the singleton $\{\mathbf{x}^*(\mathbf{y})\}$ for some $\mathbf{x}^*(\mathbf{y}) \in \mathbf{\Delta}_{\mathbf{y}}$. Then there is a measurable mapping $g : \mathbf{Y} \to \mathbf{\Delta}_{\mathbf{y}}$ such that

$$\begin{array}{rcl} g(\mathbf{y}) & = & \mathbf{x}^*(\mathbf{y}) \;\text{ for every }\; \mathbf{y} \in \mathbf{Y} \\ \rho & = & \displaystyle\int_{\mathbf{Y}} f(g(\mathbf{y}), \mathbf{y}) \, d\varphi(\mathbf{y}), \end{array} \quad (7)$$

and for every $\alpha \in \mathbb{N}^n$, and $\beta \in \mathbb{N}^p$:

$$\int_{\mathbf{\Delta}} \mathbf{x}^\alpha \mathbf{y}^\beta \, d\mu^*(\mathbf{x}, \mathbf{y}) \,=\, \int_{\mathbf{Y}} \mathbf{y}^\beta \, g(\mathbf{y})^\alpha \, d\varphi(\mathbf{y}). \quad (8)$$

(c) There is no duality gap between (5) and (6), i.e. $\rho = \rho^*$, and if $(p_i)_{i \in \mathbb{N}} \subset \mathbb{R}[\mathbf{y}]$ is a maximizing sequence of (6) then:

$$\int_{\mathbf{Y}} \mid J(\mathbf{y}) - p_i(\mathbf{y}) \mid d\varphi(\mathbf{y}) \,\to\, 0 \quad \text{as } i \to \infty. \quad (9)$$

Moreover, define the functions $(\tilde{p}_i)$ as follows: $\tilde{p}_0 := p_0$, and

$$\mathbf{y} \mapsto \tilde{p}_i(\mathbf{y}) \,:=\, \max\,[\,\tilde{p}_{i-1}(\mathbf{y}), p_i(\mathbf{y})\,], \quad i = 1, 2, \ldots$$

Then $\tilde{p}_i \to J(\cdot)$, φ-almost uniformly[1].

An optimal solution μ^* of $\mathbf{P}$ encodes *all* information on the optimal solutions $\mathbf{x}^*(\mathbf{y})$ of $\mathbf{P}_{\mathbf{y}}$. Moreover from Theorem 1(c), any optimal or nearly optimal solution of $\mathbf{P}^*$ provides us with some polynomial lower approximation of the optimal value function $\mathbf{y} \mapsto J(\mathbf{y})$ that converges to $J(\cdot)$ in the $L_1(\varphi)$ norm, and one may also obtain a piecewise polynomial approximation that converges to $J(\cdot)$, φ-almost uniformly.

[1] A sequence (g_n) on a measure space $(\mathbf{Y}, \mathcal{B}(\mathbf{Y}), \varphi)$ converges to g, φ-*almost uniformly*, if and only if for every $\epsilon > 0$, there is a set $A \in \mathcal{B}(\mathbf{Y})$ such that $\varphi(A) < \epsilon$ and $g_n \to g$, uniformly on $\mathbf{Y} \setminus A$.

In [6] the first author has defined a (J+M)-hierarchy of semidefinite relaxations $(\mathbf{Q}_i)$ to approximate as closely as desired the optimal value ρ. In particular, the dual of each semidefinite relaxation $\mathbf{Q}_i$ provides a polynomial $q_i \in \mathbb{R}[\mathbf{y}]$ bounded above by $J(\mathbf{y})$, and as $i \to \infty$, the sequence $(\tilde{q}_i)$ with $\mathbf{y} \mapsto \tilde{q}_i(\mathbf{y}) := \max_{\ell=1,\ldots i} q_\ell(\mathbf{y})$, converges φ-almost uniformly to the optimal value function J. This last property is the rationale behind the heuristic developed below.

III. A "JOINT+MARGINAL" APPROACH

Let $\mathbb{N}^n_i := \{\alpha \in \mathbb{N}^n : |\alpha| \leq i\}$ with $|\alpha| = \sum_i \alpha_i$. With a sequence $\mathbf{z} = (z_\alpha)$ indexed in the canonical basis $(\mathbf{x}^\alpha)$ of $\mathbb{R}[\mathbf{x}]$, let $L_\mathbf{z} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ be the linear mapping:

$$f\,(=\sum_\alpha f_\alpha(\mathbf{x}))\;\mapsto\; L_\mathbf{z}(f)\;:=\;\sum_\alpha f_\alpha\, z_\alpha, \qquad f \in \mathbb{R}[\mathbf{x}].$$

Moment matrix: The moment matrix $\mathbf{M}_i(\mathbf{z})$ associated with a sequence $\mathbf{z} = (z_\alpha)$, $\alpha \in \mathbb{N}^n_{2i}$, has its rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$, and with entries

$$\mathbf{M}_i(\mathbf{z})(\alpha,\beta) \;=\; L_\mathbf{z}(\mathbf{x}^{\alpha+\beta}) \;=\; z_{\alpha+\beta}, \quad \forall\, \alpha,\beta \in \mathbb{N}^n_i.$$

Localizing matrix: Let q be the polynomial $\mathbf{x} \mapsto q(\mathbf{x}) := \sum_u q_u \mathbf{x}^u$. The localizing matrix $\mathbf{M}_i(q\,\mathbf{z})$ associated with $q \in \mathbb{R}[\mathbf{x}]$ and a sequence $\mathbf{z} = (z_\alpha)$, has its rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$, and with entries.

$$\begin{aligned}\mathbf{M}_i(q\,\mathbf{z})(\alpha,\beta) &= L_\mathbf{z}(q(\mathbf{x})\mathbf{x}^{\alpha+\beta})\\ &= \sum_{u \in \mathbb{N}^n} q_u z_{\alpha+\beta+u}, \quad \forall\, \alpha,\beta \in \mathbb{N}^n_i.\end{aligned}$$

A sequence $\mathbf{z} = (z_\alpha) \subset \mathbb{R}$ is said to have a *representing* finite Borel measure supported on $\mathbf{K}$ if there exists a finite Borel measure μ such that

$$z_\alpha \;=\; \int_\mathbf{K} \mathbf{x}^\alpha\, d\mu, \qquad \forall\, \alpha \in \mathbb{N}^n.$$

A. A "joint+marginal" approach

With $\{f, (g_j)_{j=1}^m\} \subset \mathbb{R}[\mathbf{x}]$, let $\mathbf{K} \subset \mathbb{R}^n$ be the basic compact semi-algebraic set

$$\mathbf{K} := \{\mathbf{x} \in \mathbb{R}^n \,:\, g_j(\mathbf{x}) \geq 0,\; j = 1,\ldots,m\}, \tag{10}$$

and consider the polynomial optimization problem (1).

Let $\mathbf{Y}_k \subset \mathbb{R}$ be some interval $[\underline{x}_k, \overline{x}_k]$, assumed to be contained in the orthogonal projection of $\mathbf{K}$ into the x_k-ccordinate axis.

For instance when the g_j's are affine (so that $\mathbf{K}$ is a convex polytope), $\underline{x}_k$ (resp. $\overline{x}_k$) solves the linear program $\min(\text{resp}\max)\,\{x_k \,:\, \mathbf{x} \in \mathbf{K}\}$. Similarly, when $\mathbf{K}$ is convex and defined by concave polynomials, one may obtain $\underline{x}_k$ and $\overline{x}_k$, up to (arbitrary) fixed precision. In many cases, (upper and lower) bound constraints on the variables are already part of the problem definition.

Let φ_k be the probability measure uniformly distributed on $\mathbf{Y}_k$; hence its moments (β_ℓ), $\ell \in \mathbb{N}$, are given by:

$$\beta_\ell = \int_{\underline{x}_k}^{\overline{x}_k} x^\ell d\varphi_k(x) \;=\; \frac{\overline{x}_k^{\ell+1} - \underline{x}_k^{\ell+1}}{(\ell+1)(\overline{x}_k - \underline{x}_k)} \tag{11}$$

for every $\ell = 0, 1, \ldots$. Define the following parametric polynomial program in $n-1$ variables:

$$J^k(y) \;=\; \min_\mathbf{x}\,\{f(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K};\;\; x_k = y\}, \tag{12}$$

or, equivalently $J^k(y) = \min\{f(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K}_y\}$, where for every $y \in \mathbf{Y}$:

$$\mathbf{K}_y := \{\mathbf{x} \in\, \mathbf{K};\, x_k = y\}. \tag{13}$$

Observe that by definition, $f^* = \min_x\{J^k(x) \,:\, x \in \mathbf{Y}_k\}$, and $\mathbf{K}_y \neq \emptyset$ whenever $y \in \mathbf{Y}_k$, where $\mathbf{Y}_k$ is the orthogonal projection of $\mathbf{K}$ into the x_k-coordinate axis.

Semidefinite relaxations

To compute (or at least approximate) the optimal value ρ of problem $\mathbf{P}$ in (5) associated with the parametric optimization problem (12), we now provide a hierarchy of semidefinite relaxations in the spirit of those defined in [4]. Let $v_j := \lceil(\deg \mathrm{g_j})/2\rceil$, $j = 1,\ldots,m$, and for $i \geq \max_j v_j$, consider the semidefinite program:

$$\begin{aligned}\rho_{ik} = \quad & \inf_\mathbf{z} \quad L_\mathbf{z}(f) \\ & \text{s.t.} \quad \mathbf{M}_i(\mathbf{z}) \succeq 0,\; \mathbf{M}_{i-v_j}(g_j\,\mathbf{z}) \succeq 0, \quad j = 1,\ldots,m \\ & \qquad\;\; L_\mathbf{z}(x_k^\ell) = \beta_\ell, \quad \ell = 0,1,\ldots 2i,\end{aligned} \tag{14}$$

where (β_ℓ) is defined in (11). We call (14) the *parametric semidefinite relaxation* of $\mathbf{P}$ with parameter $y = x_k$. Observe that without the "moment" constraints $L_\mathbf{z}(x_k^\ell) = \beta_\ell$, $\ell = 1,\ldots 2i$, the semidefinite program (14) is a relaxation of $\mathbf{P}$ and if $\mathbf{K}$ is compact, its corresponding optimal value f_i^* converges to f^* as $k \to \infty$; see Lasserre [4].

Letting $g_0 \equiv 1$, the dual of (14) reads:

$$\begin{aligned}\rho_{ik}^* = \quad & \sup_{\lambda,(\sigma_j)} \sum_{\ell=0}^{2i} \lambda_\ell\,\beta_\ell \\ & \text{s.t.} \quad f(\mathbf{x}) - \sum_{\ell=0}^{2i} \lambda_\ell x_k^\ell = \sum_{j=0}^m \sigma_j\, g_j \\ & \qquad\;\; \sigma_j \in \Sigma[\mathbf{x}], \quad 0 \leq j \leq m; \\ & \qquad\;\; \deg \sigma_j g_j \leq 2i, \quad 0 \leq j \leq m.\end{aligned} \tag{15}$$

Equivalently, recall that $\mathbb{R}[x_k]_{2i}$ is the space of univariate polynomials of degree at most $2i$, and observe that in (15), the objective reads

$$\sum_{\ell=0}^{2i} \lambda_\ell\,\beta_\ell = \int_{\mathbf{Y}_k} p_i(y) d\varphi_k(y),$$

where $p_i \in \mathbb{R}[x_k]_{2i}$ is the univariate polynomial $x_k \mapsto p_i(x_k) := \sum_{\ell=0}^{2i} \lambda_\ell x_k^\ell$. Then equivalently, the above dual may be rewritten as:

$$\begin{aligned}\rho_{ik}^* = \quad & \sup_{p_i,(\sigma_j)} \int_{\mathbf{Y}_k} p_i d\varphi_k \\ & \text{s.t.} \quad f - p_i = \sum_{j=0}^m \sigma_j\, g_j \\ & \qquad\;\; p_i \in \mathbb{R}[x_k]_{2i};\; \sigma_j \in \Sigma[\mathbf{x}], \quad 0 \leq j \leq m; \\ & \qquad\;\; \deg \sigma_j g_j \leq 2i, \quad 0 \leq j \leq m.\end{aligned} \tag{16}$$

Assumption 1: The family of polynomials $(g_j) \subset \mathbb{R}[\mathbf{x}]$ is such that for some $M > 0$,

$$\mathbf{x} \mapsto M - \|\mathbf{x}\|^2 = \sum_{j=0}^{m} \sigma_j \, g_j,$$

for some M and some s.o.s. polynomials $(\sigma_j) \subset \Sigma[\mathbf{x}]$.

Theorem 2: Let $\mathbf{K}$ be as (10) and Assumption 1 hold. Let the interval $\mathbf{Y}_k \subset \mathbb{R}$ be the orthogonal projection of $\mathbf{K}$ into the x_k-coordinate axis, and let φ_k be the probability measure, uniformly distributed on $\mathbf{Y}_k$. Assume that $\mathbf{K}_y$ in (13) is not empty, let $y \mapsto J^k(y)$ be as in (12) and consider the semidefinite relaxations (14)-(16). Then as $i \to \infty$:

(a) $\rho_{ik} \uparrow \int_{\mathbf{Y}_k} J^k d\varphi_k$ and $\rho^*_{ik} \uparrow \int_{\mathbf{Y}_k} J^k d\varphi_k$

(b) Let $(p_i, (\sigma^i_j))$ be a nearly optimal solution of (16), e.g. such that $\int_{\mathbf{Y}_k} p_i d\varphi_k \geq \rho^*_{ik} - 1/i$. Then $p_i(y) \leq J^k(y)$ for all $y \in \mathbf{Y}_k$, and

$$\int_{\mathbf{Y}_k} |J^k(y) - p_i(y)| \, d\varphi_k(y) \;\to\; 0, \quad \text{as } i \to \infty. \tag{17}$$

Moreover, if one defines $\tilde{p}_0 := p_0$, and

$$\mathbf{y} \mapsto \tilde{p}_i(y) \;:=\; \max\left[\,\tilde{p}_{i-1}(y), p_i(y)\,\right], \quad i = 1, 2, \ldots,$$

then $\tilde{p}_i(y) \uparrow J^k(y)$, for φ_k-almost all $y \in \mathbf{Y}_k$, and so $\tilde{p}_i \to J^k$, φ_k-almost uniformly on $\mathbf{Y}_k$.

Theorem 2 is a direct consequence of [6, Corollary 2.6].

B. A "joint+marginal" algorithm for the general case

Theorem 2 provides a rationale for the following (J+M)-algorithm in the general case. In what follows we use the primal and dual semidefinite relaxations (14)-(15) with index i *fixed*.

ALGO 1: (J+M)-algorithm: non convex K, relaxation i

Set $k = 1$**;**
Step k: **Input:** $\mathbf{K}$, f, and the orthogonal projection $\mathbf{Y}_k = [\underline{x}_k, \overline{x}_k]$ of $\mathbf{K}$ into the x_k-coordinate axis, with associated probability measure φ_k, uniformly distributed on $\mathbf{Y}_k$.
Ouput: $\tilde{x}_k \in \mathbf{Y}_k$.
Solve the semidefinite program (16) and from an optimal (or nearly optimal) solution $(p_i, (\sigma_j))$ of (16), get a global minimizer $\tilde{x}_k$ of the univariate polynomial p_i on $\mathbf{Y}_k$.
If $k = n$ stop and output $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_n)$, otherwise set $k = k + 1$ and repeat.

Of course, in general the vector $\tilde{\mathbf{x}} \in \mathbb{R}^n$ does not belong to $\mathbf{K}$. Therefore a final step consists of computing a local minimum $\hat{\mathbf{x}} \in \mathbf{K}$, by using some local minimization algorithm starting with the (unfeasible) initial point $\tilde{\mathbf{x}}$. Also note that when $\mathbf{K}$ is not convex, the determination of bounds $\underline{x}_k$ and $\overline{x}_k$ for the interval $\mathbf{Y}_k$ may not be easy, and so one might be forced to use a subinterval $\mathbf{Y}'_k \subseteq \mathbf{Y}_k$ with conservative (but computable) bounds $\underline{x}'_k \geq \underline{x}_k$ and $\overline{x}'_k \leq \overline{x}_k$.

Remark 1: Theorem 2 assumes that for every $y \in \mathbf{Y}_k$, the set $\mathbf{K}_y$ in (13) is not empty, which is the case if $\mathbf{K}$ is connected. If $\mathbf{K}_y = \emptyset$ for y in some open subset of $\mathbf{Y}_k$, then the semidefinite relaxation (14) has no solution ($\rho_{ik} = +\infty$), in which case one proceeds by dichotomy on the interval $\mathbf{Y}_k$ until $\rho_{ik} < \infty$. That is, with $z := (\underline{x}_k + \overline{x}_k)/2$, consider the subintervals $\mathbf{Y}^1_k := [\underline{x}_k, z]$ and $\mathbf{Y}^2_k := [z, \overline{x}_k]$. Solve (14) where the (β_ℓ) in (11) are updated according to $\mathbf{Y}^1_k$ (resp. $\mathbf{Y}^2_k$) to obtain ρ^1_{ik} (resp. ρ^2_{ik}). If $\rho^s_{ik} < \infty$ for some s, stop and proceed with $\mathbf{Y}^s_k$ instead of $\mathbf{Y}_k$, otherwise choose randomly $\mathbf{Y}^1_k$ or $\mathbf{Y}^2_k$ and iterate.

C. A "joint+marginal" algorithm when $\mathbf{K}$ is convex

In this section, we now assume that the feasible set $\mathbf{K} \subset \mathbb{R}^n$ of problem $\mathbf{P}$ is convex (and compact). The idea is to compute $\tilde{x}_1$ as in **ALGO 1** and then repeat the procedure but now for the $(n-1)$-variable problem $\mathbf{P}(\tilde{x}_1)$ which is problem $\mathbf{P}$ in which the variable x_1 is *fixed* at the value $\tilde{x}_1$. This alternative is guaranteed to work if $\mathbf{K}$ is convex (but not always if $\mathbf{K}$ is not convex).

For every $j \geq 2$, denote by $\mathbf{x}_j \in \mathbb{R}^{n-j+1}$ the vector $(x_j, \ldots, x_n)$, and by $\tilde{\mathbf{x}}_{j-1} \in \mathbb{R}^{j-1}$ the vector $(\tilde{x}_1, \ldots, \tilde{x}_{j-1})$ (and so $\tilde{\mathbf{x}}_1 = \tilde{x}_1$).

Let the interval $\mathbf{Y}_1 \subset \mathbb{R}$ be the orthogonal projection of $\mathbf{K}$ into the x_1-coordinate axis. For every $\tilde{x}_1 \in \mathbf{Y}_1$, let the interval $\mathbf{Y}_2(\tilde{\mathbf{x}}_1) \subset \mathbb{R}$ be the orthogonal projection of the set $\mathbf{K} \cap \{\mathbf{x} : x_1 = \tilde{x}_1\}$ into the x_2-coordinate axis. Similarly, given $\tilde{\mathbf{x}}_2 \in \mathbf{Y}_1 \times \mathbf{Y}_2(\tilde{\mathbf{x}}_1)$, let the interval $\mathbf{Y}_3(\tilde{\mathbf{x}}_2) \subset \mathbb{R}$ be the orthogonal projection of the set $\mathbf{K} \cap \{\mathbf{x} : x_1 = \tilde{x}_1;\, x_2 = \tilde{x}_2\}$ into the x_3-coordinate axis, and etc. in the obvious way.

For every $k = 2, \ldots, n$, and $\tilde{\mathbf{x}}_{k-1} \in \mathbf{Y}_1 \times \mathbf{Y}_2(\tilde{\mathbf{x}}_1) \cdots \times \mathbf{Y}_{k-1}(\tilde{\mathbf{x}}_{k-2})$, let $\tilde{f}_k(\mathbf{x}_k) := f((\tilde{\mathbf{x}}_{k-1}, \mathbf{x}_k))$, and $\tilde{g}^k_j(\mathbf{x}_k) := g_j((\tilde{\mathbf{x}}_{k-1}, \mathbf{x}_k))$, $j = 1, \ldots, m$. Similarly, let

$$\begin{aligned} \mathbf{K}_k(\tilde{\mathbf{x}}_{k-1}) \;&:=\; \{\mathbf{x}_k \,:\, \tilde{g}^k_j(\mathbf{x}_k) \geq 0,\, j = 1, \ldots, m\}, \\ &=\; \{\mathbf{x}_k \,:\, (\tilde{\mathbf{x}}_{k-1}, \mathbf{x}_k) \in \mathbf{K}\}, \end{aligned} \tag{18}$$

and consider the problem:

$$\mathbf{P}(\tilde{\mathbf{x}}_{k-1}) : \quad \min\{\tilde{f}_k(\mathbf{x}_x) \,:\, \mathbf{x}_x \in \mathbf{K}_j(\tilde{\mathbf{x}}_{k-1})\}, \tag{19}$$

i.e. the original problem $\mathbf{P}$ where the variable x_ℓ is fixed at the value $\tilde{x}_\ell$, for every $\ell = 1, \ldots, k-1$.

Write $\mathbf{Y}_j(\tilde{\mathbf{x}}_{k-1}) = [\underline{x}_k, \overline{x}_k]$, and let φ_k be the probability measure uniformly distributed on $\mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})$.

Let $\mathbf{z}$ be a sequence indexed in the monomial basis of $\mathbb{R}[\mathbf{x}_k]$. With index i, fixed, the parametric semidefinite relaxation (14) with parameter x_k, associated with problem $\mathbf{P}(\tilde{\mathbf{x}}_{k-1})$, reads:

$$\begin{array}{lll} \rho_{ik} = \inf\limits_{\mathbf{z}} & L_{\mathbf{z}}(\tilde{f}_k) & \\ \quad\text{s.t.} & \mathbf{M}_i(\mathbf{z}),\, \mathbf{M}_{i-v_j}(\tilde{g}^k_j\,\mathbf{z}) \succeq 0, & j = 1, \ldots, m \\ & L_{\mathbf{z}}(x^\ell_k) = \beta_\ell, \quad \ell = 0, 1, \ldots, 2i, & \end{array} \tag{20}$$

where (β_ℓ) is defined in (11). Its dual is the semidefinite program (with $\tilde{g}^k_0 \equiv 1$)):

$$\begin{array}{rcl} \rho^*_{ik} &=& \sup\limits_{p_i, (\sigma_j)} \displaystyle\int_{\mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})} p_i d\varphi_k \\ && \text{s.t.} \quad \tilde{f}_k - p_i = \sigma_0 + \displaystyle\sum_{j=1}^{m} \sigma_j\, \tilde{g}^k_j \\ && \qquad p_i \in \mathbb{R}[x_k]_{2i},\; \sigma_j \in \Sigma[\mathbf{x}_k], \quad j = 0, \ldots, m \\ && \qquad \deg \sigma_j \tilde{g}^k_j \leq 2i, \quad j = 0, \ldots, m. \end{array} \tag{21}$$

The important difference between (14) and (20) is the *size* of the corresponding semidefinite programs, since $\mathbf{z}$ in (14) (resp. in (20)) is indexed in the canonical basis of $\mathbb{R}[\mathbf{x}]$ (resp. $\mathbb{R}[\mathbf{x}_k]$).

The (J+M)-algorithm for $\mathbf{K}$ *convex*

Recall that the order i of the semidefinite relaxation is fxed. The (J+M)-algorithm consists of n steps. At step k of the algorithm, the vector $\tilde{\mathbf{x}}_{k-1} = (\tilde{x}_1, \ldots, \tilde{x}_{k-1})$ (already computed) is such that $\tilde{x}_1 \in \mathbf{Y}_1$ and $\tilde{x}_\ell \in \mathbf{Y}_\ell(\tilde{\mathbf{x}}_{\ell-1})$ for every $\ell = 2, \ldots, k-1$, and so the set $\mathbf{K}_k(\tilde{\mathbf{x}}_{k-1})$ is a nonempty compact convex set.

ALGO 2: (J+M)-algorithm: convex K, relaxation i

```
**Set** $k = 1$;
**Step** $k \geq 1$: **Input:** For $k = 1$, $\tilde{\mathbf{x}}_0 = \emptyset$, $\mathbf{Y}_1(\tilde{\mathbf{x}}_0) = \mathbf{Y}_1$; $\mathbf{P}(\tilde{\mathbf{x}}_0) = \mathbf{P}$, $f_1 = f$ and $\tilde{g}_j^1 = g_j$, $j = 1, \ldots, m$.
For $k \geq 2$, $\tilde{\mathbf{x}}_{k-1} \in \mathbf{Y}_1 \times \mathbf{Y}_2(\tilde{x}_1) \cdots \times \mathbf{Y}_{k-1}(\tilde{x}_{k-2})$.
**Output:** $\tilde{\mathbf{x}}_k = (\tilde{\mathbf{x}}_{k-1}, \tilde{x}_k)$ with $\tilde{x}_k \in \mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})$.
Consider the parametric semidefinite relaxations (20)-(21) with parameter $x_k$, associated with problem $\mathbf{P}(\tilde{\mathbf{x}}_{k-1})$ in (19).
  • From an optimal solution of (21), extract the univariate polynomial $x_k \mapsto p_i(x_k) := \sum_{\ell=0}^{2i} \lambda_\ell^* x_k^\ell$.
  • Get a global minimizer $\tilde{x}_k$ of $p_i$ on the interval $\mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1}) = [\underline{x}_k, \overline{x}_k]$, and set $\tilde{\mathbf{x}}_k := (\tilde{\mathbf{x}}_{k-1}, \tilde{x}_k)$.
If $k = n$ stop and ouput $\tilde{\mathbf{x}} \in \mathbf{K}$, otherwise set $k = k + 1$ and repeat.
```

As $\mathbf{K}$ is convex, $\tilde{\mathbf{x}} \in \mathbf{K}$ and one may stop. A refinement is to now use $\tilde{\mathbf{x}}$ as the initial guess of a local minimization algorithm to obtain a local minimizer $\hat{\mathbf{x}} \in \mathbf{K}$ of $\mathbf{P}$. In view of Theorem 2, the larger the index i of the relaxations (20)-(21), the better the values $f(\tilde{\mathbf{x}})$ and $f(\hat{\mathbf{x}})$.

Of course, **ALGO 2** can also be used when $\mathbf{K}$ is not convex. However, it may happen that at some stage k, the semidefinite relaxation (20) may be infeasible because $J^k(y)$ is infinite for some values of $y \in \mathbf{Y}_k(\tilde{\mathbf{x}}_{k-1})$. This is because the feasible set $\mathbf{K}(\tilde{\mathbf{x}}_{k-1})$ in (18) may be disconnected.

IV. Computational experiments

We report on preliminary computational experiments on some non convex NP-hard optimization problems. We have tested the algorithms on a set of difficult global optimization problems taken from Floudas et al. [1]. To solve the semidefinite programs involved in **ALGO 1** and in **ALGO 2**, we have used the GloptiPoly software [3] that implements the hierarchy of semidefinite relaxations defined in [4, (4.5)].

A. **ALGO 2** *for convex set* $\mathbf{K}$

Those problems are taken from [1, §2]. The set $\mathbf{K}$ is a convex polytope and the function f is a nonconvex quadratic polynomial $\mathbf{x} \mapsto \mathbf{x}'Q\mathbf{x} + \mathbf{b}'\mathbf{x}$ for some real symmetric matrix Q and vector $\mathbf{b}$. In Table I one displays the problem name, the number n of variables, the number m of constraints, the gobal optimum f^*, the index i of the semidefinite relaxation in **ALGO 2**, the optimal value obtained using the output of **ALGO 2** as initial guess in a local minimization algorithm of the MATLAB toolbox, and the associated relative error.

Prob	n	m	f^*	i	ALGO 2	rel. error
2.2	5	11	-17	2	-17.00	0%
2.3	6	8	-361.5	1	-361.50	0%
2.6	10	21	-268.01	1	-267.00	0.3%
2.9	10	21	0	1	0.00	0%
2.8C1	20	30	-394.75	1	-385.30	2.4%
2.8C2	20	30	-884.75	1	-871.52	1.5%
2.8C3	20	30	-8695	1	-8681.7	0.15%
2.8C4	20	30	-754.75	1	-754.08	0.09%
2.8C5	20	30	-4150.41	1	-3678.2	11%

TABLE I
ALGO 2 FOR CONVEX SET $\mathbf{K}$

As recommended in GloptiPoly [3] for numerical stability and precision, the problem data have been rescaled to obtain a polytope contained in the box $[-1, 1]^n$. As one may see, and excepted for problem 2.8C5, the relative error is very small. For the last problem the relative error (about 11%) is relatively high despite enforcing some extra upper and lower bounds $\underline{x_i} \leq x_i \leq \overline{x}_i$, after reading the optimal solution. However, using $\tilde{\mathbf{x}} \in \mathbf{K}$ as initial guess of the local minimization algorithm in MATLAB, one still finds the optimal value f^*.

B. **ALGO 1** *for non convex set* $\mathbf{K}$

Again in Table II below, n (resp. m) stands for the number of variables (resp. constraints), and the value displayed in the "**ALGO 1**" column is obtained in running a local minimization algorithm of the MATLAB toolbox with the output $\tilde{\mathbf{x}}$ of **ALGO 1** as initial guess.

In Problems 3.2, 3.3 and 3.4 from Floudas et al. [1, §3], one has $2n$ linear bound constraints and additional linear and non convex quadratic constraints. As one may see, the results displayed in Table II are very good.

For the Haverly Pooling problem 5.2.2 in [1, §5] with three different data sets, one has $n = 9$ and $m = 24$ constraints, among which 3 nonconvex bilinear constraints and 18 linear bound constraints $0 \leq x_i \leq 500$, $i = 1, \ldots, 9$. In the first run of **ALGO 1** we obtained bad results because the bounds are very loose and in the hierarchy of lower bounds (f_k^*) in [4] that converge to f^*, if on the one hand $f_2^* = f^*$, on the other hand the lower bound $f_1^* < f^*$ is loose. In such a case, and in view of the rationale behind the "joint+marginal" approach, it is illusory to obtain good results with **ALGO 1** or **ALGO 2**. Therefore, from the optimal solution $\mathbf{x}^*$ in [1], and when $0 < x_i^* < 500$, we have generated stronger bounds $0.4x_i^* \leq x_i \leq 1.6x_i^*$. In this case, f_1^* is much closer to f^* and we obtain the global minimum f^* with **ALGO 1** followed by the local minimization subroutine; see Table II. Importantly, in **ALGO 1**, and before running the local optimization subroutine, one ends up with a non feasible point $\tilde{\mathbf{x}}$. Moreover, we had to sometimes use the dichotomy procedure of Remark 1 because if $\mathbf{Y}_k$ is large, one may have $\mathbf{K}_y = \emptyset$ for y in some open subintervals of $\mathbf{Y}_k$.

Problem 7.2.2 has 13 linear constraints and 4 nonlinear constraints with bilinear terms. To handle the non-polynomial

Prob	n	m	f^*	i	ALGO 1	rel. error
3.2	8	22	7049	1	7049	0%
3.3	5	16	-30665	1	-30665	0%
3.4	6	18	-310	1	-310	0%
5.2.2 (1)	9	24	400	1	400	0%
5.2.2 (2)	9	24	600	1	600	0%
5.2.3 (3)	9	24	750	1	750	0%
5.2.4	9	24	450	1	450	0%
7.2.2	6	17	-0.3746	1	-0.3746	0%
7.2.3	8	22	7049.25	1	7049.25	0%
7.2.5	5	16	10122	1	10122	0%
7.2.6	3	7	-83.254	1	-83.258	1%
8.2.8	6	17	-0.3768	1	-0.3767	0%

TABLE II

ALGO 1 FOR NON CONVEX SET $\mathbf{K}$

n	20	30	40
$(\rho - f_1^*)/\lvert f_1^*\rvert$	10.3%	12.3%	12.5%

TABLE III

RELATIVE ERROR FOR MAXCUT

function $x_i^{0.5}$, one uses the lifting $u_i^2 = x_i$, $u_i \geq 0$, $i = 5, 6$. Problem 7.2.6 has only 3 variables, 6 linear bound constraints, and one highly nonlinear constraint (and criterion). Here one uses the lifting $u\,x_2 = 1$, to handle the term x_2^{-1}. Again one obtains the optimal value f^* with **ALGO 1** followed by a local optimization subroutine.

Notice that in some examples, running the local procedure with 100 randomly chosen initial guesses, also gave the global optimum with a high percentage of success. On the other hand, the rate of success was less than 17% for Prob 3.4, 10% for Prob. 5.2.2(2), 6% for Prob 5.2.4, 35% for Prob. 8.2.8, and 60% for Probs 5.2.2(1), 5.2.2(3), 7.2.6.

C. **ALGO 2** *for MAXCUT*

Finally we have tested **ALGO 2** on the famous NP-hard discrete optimization problem MAXCUT, which consists of minimizing a quadratic form $\mathbf{x} \mapsto \mathbf{x}'Q\mathbf{x}$ on $\{-1, 1\}^n$, for some real symmetric matrix $Q \in \mathbb{R}^{n \times n}$. In this case, $\mathbf{Y}_k = \{-1, 1\}$ and the marginal constraint $L_{\mathbf{z}}(x_k^\ell) = \gamma_\ell$ in (20) need only be imposed for $\ell = 1$, because of the constraints $x_k^2 = 1$ for every $k = 1, \ldots, n$. Accordingly, in an optimal solution of the dual (21), $p_i \in \mathbb{R}[x_k]$ is an affine polynomial $x_k \mapsto p_i(x_k) = \lambda_0 + \lambda_1 x_k$ for some scalars λ_0, λ_1. Therefore after solving (21) one decides $\tilde{x}_k = -1$ if $p_i(-1) < p_i(1)$ (i.e. if $\lambda_1 > 0$) and $\tilde{x}_k = 1$ otherwise.

Recall that in **ALGO 2** one first compute $\tilde{x}_1$, then with x_1 fixed at the value $\tilde{x}_1$, one computes $\tilde{x}_2$, etc. until one finally computes $\tilde{x}_n$, and get $\tilde{\mathbf{x}}$. In what we call the "max-gap" variant of **ALGO 2**, one first solves n programs (14)-(15) with parameter x_1 to obtain an optimal solution $p_i(x_1) = \lambda_0^1 + \lambda_1^1 x_1$ of the dual (15), then with x_2 to obtain $(\lambda_0^2, \lambda_1^2)$, etc., and finally with x_n to obtain $(\lambda_0^n, \lambda_1^n)$. One then select k such that $|\lambda_1^k| = \max_\ell |\lambda_1^\ell|$, and compute $\tilde{x}_k$ accordingly. This is because the larger $|\lambda_1|$, (i.e. the larger $|p_i(-1) - p_i(1)|$), the more likely the choice -1 or 1 is correct. After x_k is fixed at the value $\tilde{x}_k$, one repeats the procedure for the $(n-1)$-problem $\mathbf{P}(\tilde{x}_k)$, etc.

We have tested the "max-gap" variant for MAXCUT problems on random graphs with $n = 20, 30$ and 40 nodes. For each value of n, we have solved 50 randomly generated problems and 100 for $n = 40$. The probability φ_k on $\mathbf{Y}_k = \{-1, 1\}$ is uniform (i.e., $\beta_1 = 0$ in (20)). Let f_1^* denote the optimal value of the Shor's relaxation with Goemans and Williamson's 0.878 performance guarantee. Let ρ denote the cost of the solution $\mathbf{x} \in \{-1, 1\}^n$ generated by the **ALGO 2**. In Table III we have reported the average relative error $(\rho - f_1^*)/|f_1^*|$, which as one may see, is comparable with the Goemans and Williamson (GW) ratio. (Recall that the relative error is measured with respect to the lower bound f_1^* and *not* the optimal value f^*.)

V. CONCLUSION

First preliminary results are promising, even with small relaxation order i. In general, the efficiency of **ALGO 1** or **ALGO 2** with i fixed, should be related to how close to the global optimum f^* is the optimal value f_i^* at step i of the hierarchy of relaxations defined in [4] to approximate f^*. When the feasible set is non convex, it may become difficult to obtain a feasible solution and an interesting issue for further investigation is how to proceed when $\mathbf{K}_y = \emptyset$ for y in some open subinterval of $\mathbf{Y}_k$ (proceeding by dichotomy on $\mathbf{Y}_k$ is one possiblity).

REFERENCES

[1] C.A. Floudas et al., *Handbook of Test Problems in Local and Global optimization*, Kluwer Academic Publishers, Dordrecht, 1999.

[2] M.X. Goemans, D.P. Williamson, *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*, Journal of the ACM 42, pp.1115-1145, 1995.

[3] D. Henrion, J. B. Lasserre, J. Lofberg, *GloptiPoly 3: moments, optimization and semidefinite programming*, Optim. Methods and Softw. 24, pp. 761–779, 2009. `http://www.laas.fr/~henrion/software/gloptipoly3/`

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

[5] J.B. Lasserre, *Polynomial programming: LP-relaxations also converge*, SIAM J. Optim. 15, pp. 383–393, 2004.

[6] J.B. Lasserre, *A "joint+marginal" approach to parametric polynomial optimization*, SIAM J. Optim. 20, pp. 1995-2022, 2010.

Appendix C

Paper: “A “joint + marginal" heuristic for 0/1 programs"

J Glob Optim
DOI 10.1007/s10898-011-9788-9

A “joint + marginal” heuristic for 0/1 programs

Jean B. Lasserre · Tung Phan Thanh

Received: 24 February 2011 / Accepted: 15 September 2011

Abstract We propose a heuristic for 0/1 programs based on the recent “joint + marginal” approach of the first author for parametric polynomial optimization. The idea is to first consider the n-variable $(x_1, \ldots, x_n)$ problem as a $(n-1)$-variable problem $(x_2, \ldots, x_n)$ with the variable x_1 being now a parameter taking value in $\{0, 1\}$. One then solves a hierarchy of what we call “joint + marginal” semidefinite relaxations whose duals provide a sequence of polynomial approximations $x_1 \mapsto J_k(x_1)$ that converges to the optimal value function $J(x_1)$ (as a function of the parameter x_1). One considers a fixed index k in the hierarchy and if $J_k(1) > J_k(0)$ then one decides $x_1 = 1$ and $x_1 = 0$ otherwise. The quality of the approximation depends on how large k can be chosen (in general, for significant size problems, $k = 1$ is the only choice). One iterates the procedure with now a $(n-2)$-variable problem with one parameter $x_2 \in \{0, 1\}$, etc. Variants are also briefly described as well as some preliminary numerical experiments on the MAXCUT, k-cluster and 0/1 knapsack problems.

Keywords 0/1 Programs · Semidefinite relaxations

1 Introduction

Consider the general 0/1 program

$$\mathbf{P}: \quad f^* = \min_{\mathbf{x}} \{ f(\mathbf{x}) \ : \ \mathbf{x} \in \mathbf{K} \cap \{0, 1\}^n \}$$

where f is a polynomial and $\mathbf{K} \subset \mathbb{R}^n$ is a basic closed semi-algebraic set. One way to approximate the optimal value of $\mathbf{P}$ is to solve a hierarchy of either LP-relaxations as in

J. B. Lasserre ( ) · T. P. Thanh
LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse Cédex 4, France
e-mail: lasserre@laas.fr

T. P. Thanh
e-mail: tphantha@laas.fr

J. B. Lasserre
LAAS and Institute of Mathematics, LAAS, 7 Avenue du Colonel Roche, 31077 Toulouse Cédex 4, France

Published online: 27 September 2011

Sherali–Adams [14] and Lovász–Schrijver [11], or semidefinite (SDP) relaxations as in Lovász–Schrijver [11] or Lasserre [6] (the latter being defined for general polynomial optimization problems). For 0/1 programs, the convergence of both LP and semidefinite relaxations of Lasserre [6], Lovász–Schrijver [11] and Sherali–Adams [14] is finite and for the semidefinite relaxations [6], there is a stopping criterion which, when met, guarantees that the semidefinite relaxation is exact and one may extract global minimizers. If on the one hand SDP relaxations are more powerful than LP relaxations, on the other hand the present status of SDP solvers is far from that of LP solvers in terms of size of problems that can be handled. For a comparison of those approches for 0/1 programs, the interested reader is referred to e.g. Lasserre [7] and Laurent [10].

Despite that practice seems to reveal that the finite convergence is fast, the matrix size in the kth semidefinite relaxation of the hierarchy grows up as fast as $O(n^k)$. Hence for problems of reasonable size, and in view of the present status of SDP solvers, one can implement only the first or second relaxation, which in general only provides a lower bound on f^*. Of course, this lower bound can be exploited in some other search procedure, like e.g. Branch & Bound, but more generally, the following natural question arises:

How can we use the results of the kth SDP relaxation to find (or help find) an approximate solution of the original problem?

In some well-known special cases like e.g. the MAXCUT problem, one may generate a feasible solution with guaranteed performance, e.g. from the Goemans and Williamson randomized rounding procedure [3] that uses an optimal solution of the first SDP relaxation (i.e. with $k = 1$). But there is no recipe for the general case and one is left with the possibility to use the lower bound provided by the SDP relaxation in a standard Branch & Bound.

Contribution. We here provide a simple heuristic for 0/1 polynomial programs which builds upon the so-called “joint + marginal” approach (in short (J + M)) recently developed in Lasserre [8] for *parametric* polynomial optimization. The (J + M)-approach for polynomial optimization problems with variables $\mathbf{x} \in \mathbb{R}^n$ and parameters $\mathbf{y}$ in a simple set $\mathbf{Y}$, consists of the standard hierarchy of SDP relaxations in Lasserre [5,6] where one treats the parameters $\mathbf{y}$ also as variables but now with the additional constraint that some marginal distribution on $\mathbf{Y}$ (e.g. the uniform probability distribution on $\mathbf{Y}$) is fixed. Among other things, it permits to provide a polynomial approximation of the optimal value function $\mathbf{y} \mapsto J(\mathbf{y})$ (viewed as a function of the parameter). For more details, the interested reader is referred to Lasserre [8].

In the context of a non-parametric 0/1 polynomial optimization, the above (J + M)-approach with (fixed) parameter k can be used as follows:

- (a) Treat x_1 as a parameter in $\mathbf{Y} := \{0, 1\}$ with distribution $(p_1, 1 - p_1)$ for some given $0 < p_1 < 1$, fixed arbitrary (typically $p_1 = 1/2$).
- (b) solve the (single) kth SDP relaxation of the (J + M)-hierarchy applied to problem $\mathbf{P}$ with $n - 1$ variables $x_2, \ldots, x_n$ and parameter $x_1 \in \{0, 1\}$. The dual provides a polynomial map $x_1 \mapsto J_k(x_1)$. Here k is fixed but if k increases then $J_k(x_1)$ converges to $J(x_1)$. (The map $v \mapsto J(v)$ denotes the optimal value function of $\mathbf{P}$ given that the variable x_1 is fixed at the value v). Therefore, to decide if $x_1 = 0$ or 1 in an optimal solution, one replaces the exact test $J(1) > J(0)$ with the approximate test $J_k(1) > J_k(0)$, and of course, the larger k the better; in fact the latter test becomes exact for k sufficiently large.
- (c) If $J_k(1) < J_k(0)$ then fix $\tilde{x}_1 := 1$ (and $\tilde{x}_1 = 0$ otherwise). For feasibility, check if there exists $\mathbf{x} \in \mathbf{K} \cap \{0, 1\}^n$ with $x_1 = \tilde{x}_1$; if not then set $x_1 = 1 - \tilde{x}_1$. For general 0/1 programs this step is NP-hard. However, when k is large enough, feasibility is guaranteed because $J_k(\tilde{x}_1) \approx J(\tilde{x}_1)$ implies that there is an optimal solution $\mathbf{x} \in \mathbf{K} \cap \{0, 1\}^n$ with $x_1 = \tilde{x}_1$.

Moreover, for several interesting problems like MAXCUT, k-cluster, 0/1-knapsack, the feasibility issue is trivial.

- (d) Iterate and go to step (a) with now a 0/1 program $\mathbf{P}(\tilde{x}_1)$ with $n-2$ variables $x_3, \ldots, x_n$ and parameter x_2 in $\{0, 1\}$ with distribution $(p_2, 1-p_2)$, and with $0 < p_2 < 1$ arbitrary. Of course, all data of $\mathbf{P}(\tilde{x}_1)$ are updated according to the value 1 or 0 taken by $\tilde{x}_1$.

After n iterations, one ends up with a feasible solution $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_n)$. The computational cost is less than solving n times the kth SDP relaxation in the (J + M)-hierarchy, which is itself of same order than the kth SDP relaxation in the hierarchy defined in Lasserre [6] (in fact, when $k = 1$ it includes only one additional constraint!).

Of course, in step (b) one may instead use the kth (or higher order) LP relaxation in view of the size limitation of current SDP solvers. An alternative to steps (a)–(b)–(c) is to proceed like in standard Branch & Bound. That is, compute the value δ_{0k} (resp. δ_{1k}) of kth SDP relaxation associated with $\mathbf{P}$ and the additional constraint $x_1 = 0$ (resp. $x_1 = 1$), and branch on $x_1 = 0$ if $\delta_{0k} < \delta_{1k}$, and on $x_1 = 1$ otherwise; then iterate with x_2, etc. But this requires to solve *two* SDP relaxations of same size (with only one linear constraint less) instead of one in the parametric approach. Moreover, the parametric approach can easily deal with *groups* of variables (rather than 1 variable) at a time. Indeed, with $s \in \mathbb{N}$ fixed, consider $(x_1, \ldots, x_s) \in \mathbf{Y} := \{0, 1\}^s$ with associated probability distribution φ uniformly distributed on $\mathbf{Y}$. Then in step (b) one now solves the kth SDP relaxation of the (J + M)-hierarchy applied to problem $\mathbf{P}$ with $n - s$ variables $x_{s+1}, \ldots, x_n$ and parameter $(x_1, \ldots, x_s) \in \{0, 1\}^s$. The dual provides a (square free) polynomial map $(x_1, \ldots, x_s) \mapsto J_k(x_1, \ldots, x_s)$ that converges to $J(x_1, \ldots, x_s)$ as k increases. Then one selects a global minimizer of J_k on $\mathbf{Y}$ by inspecting 2^s values, and one iterates the procedure with now a 0/1 problem with $n - s$ variables, etc.

Compared with the standard kth SDP relaxation, the kth SDP relaxation of the $(J + M)$-hierarchy has only n_k^s additional linear equality constraints where n_k^s counts all monomials $x_1^{\beta_1} \cdots x_s^{\beta_s}$ of degree at most $2k$, with $\beta_i \leq 1$ for all $i = 1, \ldots, s$.

Concerning computational experiments, we have tested the basic version with $k = 1$ on a sample of MAXCUT, k-cluster and 0/1 knapsack problems with up to 50 and 80 variables and we have compared the cost of our solution with the lower bound obtained with the first (now standard) SDP-relaxation. In average, we have obtained small relative errors of at most 5.7% on k-cluster, 2% on knapsack and 3.5% on MAXCUT. For the latter MAXCUT problems, we are close by less than 1% to the Goemans-Williamson solution obtained via the randomized rounding procedure described in Goemans and Williamson [3]. In evaluating those results, the reader should keep in mind that the "joint + marginal" heuristic is relatively general as opposed to an ad hoc algorithm that takes into account specific features of the given problem to solve.

2 The "joint + marginal" approach for parametric optimization

Most of the material in this section is taken from Lasserre [8]. Let $\mathbb{R}[\mathbf{x}, \mathbf{y}]$ denote the ring of polynomials in the variables $\mathbf{x} = (x_1, \ldots, x_n)$, and the variables $\mathbf{y} = (y_1, \ldots, y_p)$, whereas $\mathbb{R}[\mathbf{x}, \mathbf{y}]_k$ denotes its subspace of polynomials of degree at most k. Let $\Sigma[\mathbf{x}, \mathbf{y}] \subset \mathbb{R}[\mathbf{x}, \mathbf{y}]$ denote the subset of polynomials that are sums of squares (in short s.o.s.). For a real symmetric matrix $\mathbf{A}$ the notation $\mathbf{A} \succeq 0$ stands for $\mathbf{A}$ is positive semidefinite.

The parametric optimization problem

Let $\mathbf{Y} \subset \mathbb{R}^p$ be a compact set, called the *parameter* set, and for each $\mathbf{y} \in \mathbf{Y}$, fixed, consider the following *parametric* optimization problem:

$$\mathbf{P}_{\mathbf{y}}: \qquad J(\mathbf{y}) := \inf_{\mathbf{x}} \{ f(\mathbf{x}, \mathbf{y}) \,:\, h_j(\mathbf{x}, \mathbf{y}) \geq 0, \; j = 1, \ldots, m \} \tag{2.1}$$

for some polynomials $f, h_j \in \mathbb{R}[\mathbf{x}, \mathbf{y}]$, $j = 1, \ldots, m$.

The interpretation is as follows: $\mathbf{Y}$ is a set of parameters and for each instance $\mathbf{y} \in \mathbf{Y}$ of the parameter, one wishes to compute an optimal *decision* vector $\mathbf{x}^*(\mathbf{y})$ that solves problem (2.1). Let φ be a Borel probability measure on $\mathbf{Y}$, with a positive density with respect to the Lebesgue measure on $\mathbb{R}^p$ (or with respect to the counting measure if $\mathbf{Y}$ is discrete). For instance

$$\varphi(B) := \left(\int_{\mathbf{Y}} d\mathbf{y} \right)^{-1} \int_{\mathbf{Y} \cap B} d\mathbf{y}, \quad \forall B \in \mathcal{B}(\mathbb{R}^p),$$

is uniformly distributed on $\mathbf{Y}$. Sometimes, e.g. in the context of optimization with data uncertainty, φ is already specified.

The idea is to use φ (or more precisely, its moments) to get information on the mapping $\mathbf{y} \mapsto J(\mathbf{y})$ and on the distribution of optimal solutions $\mathbf{x}^*(\mathbf{y})$ of $\mathbf{P}_{\mathbf{y}}$, viewed as random vectors.

A related infinite-dimensional linear program

Let $\mathbf{K} \subset \mathbb{R}^n \times \mathbb{R}^p$ be the set:

$$\mathbf{K} := \{ (\mathbf{x}, \mathbf{y}) \,:\, \mathbf{y} \in \mathbf{Y}; \quad h_j(\mathbf{x}, \mathbf{y}) \geq 0, \quad j = 1, \ldots, m \}, \tag{2.2}$$

and for each $\mathbf{y} \in \mathbf{Y}$, let

$$\mathbf{K}_{\mathbf{y}} := \{ \mathbf{x} \in \mathbb{R}^n \,:\, h_j(\mathbf{x}, \mathbf{y}) \geq 0, \quad j = 1, \ldots, m \}. \tag{2.3}$$

In what follows we assume that for every $\mathbf{y} \in \mathbf{Y}$, the set $\mathbf{K}_{\mathbf{y}}$ in (2.3) is nonempty.

Let $\mathbf{M}(\mathbf{K})$ be the set of finite Borel probability measures on $\mathbf{K}$, and consider the following infinite-dimensional linear program $\mathbf{P}$:

$$\rho := \inf_{\mu \in \mathbf{M}(\mathbf{K})} \left\{ \int_{\mathbf{K}} f \, d\mu \,:\, \pi\mu = \varphi \right\} \tag{2.4}$$

where $\pi\mu$ denotes the marginal of μ on $\mathbb{R}^p$, that is, $\pi\mu$ is a probability measure on $\mathbb{R}^p$ defined by $\pi\mu(B) := \mu(\mathbb{R}^n \times B)$ for all $B \in \mathcal{B}(\mathbb{R}^p)$. Notice that $\mu(\mathbf{K}) = 1$ for any feasible solution μ of $\mathbf{P}$. Indeed, as φ is a probability measure and $\pi\mu = \varphi$ one has $1 = \varphi(\mathbf{Y}) = \mu(\mathbb{R}^n \times \mathbb{R}^p) = \mu(\mathbf{K})$.

The dual of $\mathbf{P}$ is the the following infinite-dimensional linear program:

$$\begin{aligned} \rho^* := \sup_{p \in \mathbb{R}[\mathbf{y}]} & \int_{\mathbf{Y}} p(\mathbf{y}) \, d\varphi(\mathbf{y}) \\ & f(\mathbf{x}, \mathbf{y}) - p(\mathbf{y}) \geq 0 \quad \forall (\mathbf{x}, \mathbf{y}) \in \mathbf{K}. \end{aligned} \tag{2.5}$$

Recall that a sequence of measurable functions (g_n) on a measure space $(\mathbf{Y}, \mathcal{B}(\mathbf{Y}), \varphi)$ converges to g *φ-almost uniformly* if and only if for every $\epsilon > 0$, there is a set $A \in \mathcal{B}(\mathbf{Y})$ such that $\varphi(A) < \epsilon$ and $g_n \to g$ uniformly on $\mathbf{Y} \setminus A$.

Theorem 2.1 ([8]) *Let both* $\mathbf{Y} \subset \mathbb{R}^p$ *and* $\mathbf{K}$ *in* (2.2) *be compact and assume that for every* $\mathbf{y} \in \mathbf{Y}$*, the set* $\mathbf{K_y} \subset \mathbb{R}^n$ *in* (2.3) *is nonempty. Let* $\mathbf{P}$ *be the optimization problem* (2.4) *and let* $\mathbf{X}^*_{\mathbf{y}} := \{\mathbf{x} \in \mathbb{R}^n \,:\, f(\mathbf{x}, \mathbf{y}) = J(\mathbf{y})\}, \mathbf{y} \in \mathbf{Y}$*. Then:*

(a) $\rho = \int_{\mathbf{Y}} J(\mathbf{y})\, d\varphi(\mathbf{y})$ *and* $\mathbf{P}$ *has an optimal solution.*

(b) *Assume that for* φ*-almost* $\mathbf{y} \in \mathbf{Y}$*, the set of minimizers of* $\mathbf{X}^*_{\mathbf{y}}$ *is the singleton* $\{\mathbf{x}^*(\mathbf{y})\}$ *for some* $\mathbf{x}^*(\mathbf{y}) \in \mathbf{K}_y$*. Then there is a measurable mapping* $g : \mathbf{Y} \to \mathbf{K_y}$ *such that*

$$g(\mathbf{y}) = \mathbf{x}^*(\mathbf{y}) \;\; \textit{for every} \;\; \mathbf{y} \in \mathbf{Y}\,; \quad \rho = \int_{\mathbf{Y}} f(g(\mathbf{y}), \mathbf{y})\, d\varphi(\mathbf{y}), \tag{2.6}$$

and for every $\alpha \in \mathbb{N}^n$*, and* $\beta \in \mathbb{N}^p$*:*

$$\int_{\mathbf{K}} \mathbf{x}^\alpha \mathbf{y}^\beta \, d\mu^*(\mathbf{x}, \mathbf{y}) = \int_{\mathbf{Y}} \mathbf{y}^\beta \, g(\mathbf{y})^\alpha \, d\varphi(\mathbf{y}). \tag{2.7}$$

(c) *There is no duality gap between* (2.4) *and* (2.5)*, i.e.* $\rho = \rho^*$*, and if* $(p_i)_{i \in \mathbb{N}} \subset \mathbb{R}[\mathbf{y}]$ *is a maximizing sequence of* (2.5) *then:*

$$\int_{\mathbf{Y}} |\, J(\mathbf{y}) - p_i(\mathbf{y})\,|\, d\varphi(\mathbf{y}) \;\to\; 0 \quad \textit{as } i \to \infty. \tag{2.8}$$

Moreover, define the functions $(\tilde{p}_i)$ *as follows:*

$$\tilde{p}_0 := p_0, \quad \mathbf{y} \mapsto \tilde{p}_i(\mathbf{y}) := \max\left[\, \tilde{p}_{i-1}(\mathbf{y}), p_i(\mathbf{y})\,\right], \quad i = 1, 2, \ldots$$

Then $\tilde{p}_i \to J(\cdot)\varphi$*-almost uniformly.*

An optimal solution μ^* of $\mathbf{P}$ encodes *all* information on the optimal solutions $\mathbf{x}^*(\mathbf{y})$ of $\mathbf{P_y}$. For instance, let $\mathbf{B}$ be a given Borel set of $\mathbb{R}^n$. Then from Theorem 2.1,

$$\text{Prob}\,(\mathbf{x}^*(\mathbf{y}) \in \mathbf{B}) = \mu^*(\mathbf{B} \times \mathbb{R}^p) = \int_{\mathbf{Y}} \mathrm{I}_B(g(\mathbf{y}))\, d\varphi(\mathbf{y}) = \varphi[g^{-1}(B) \cap \mathbf{Y}],$$

with g as in Theorem 2.1(b).

Moreover from Theorem 2.1(c), any optimal or nearly optimal solution of $\mathbf{P}^*$ provides us with some polynomial lower approximation (p_i) of the optimal value function $\mathbf{y} \mapsto J(\mathbf{y})$ that converges to $J(\cdot)$ in the $L_1(\varphi)$ norm. Moreover, one may also obtain a piecewise polynomial approximation $(\tilde{p}_i)$ that converges to $J(\cdot)$, φ-almost uniformly.

In [8] the first author has defined a (J + M)-hierarchy of SDP relaxations $(\mathbf{Q}_k)$ to approximate as closely as desired the optimal value ρ. In particular, the dual of each SDP relaxation $\mathbf{Q}_k$ provides a polynomial $p_k \in \mathbb{R}[\mathbf{y}]$ bounded above by $J(\mathbf{y})$, and $\mathbf{y} \mapsto \tilde{p}_k(\mathbf{y}) := \max_{\ell=1,\ldots k} p_k(\mathbf{y})$ converges φ-almost uniformly to the optimal value function J. This last property is the rationale behind the heuristic for polynomial 0/1 programs developed below.

3 A "joint + marginal" heuristic for 0/1 polynomial programs

Let $\mathbb{N}^n_i := \{\alpha \in \mathbb{N}^n : |\alpha| \le i\}$ with $|\alpha| = \sum_i \alpha_i$. With a sequence $\mathbf{z} = (z_\alpha)$, $\alpha \in \mathbb{N}^n$, indexed in the canonical basis $(\mathbf{x}^\alpha)$ of $\mathbb{R}[\mathbf{x}]$, let $L_\mathbf{z} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ be the linear mapping:

$$f\left(= \sum_\alpha f_\alpha(\mathbf{x})\right) \mapsto L_\mathbf{z}(f) := \sum_\alpha f_\alpha z_\alpha, \quad f \in \mathbb{R}[\mathbf{x}].$$

Moment matrix

The moment matrix $\mathbf{M}_i(\mathbf{z})$ associated with a sequence $\mathbf{z} = (z_\alpha)$, has its rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$, and with entries:

$$\mathbf{M}_i(\mathbf{z})(\alpha, \beta) = L_{\mathbf{z}}(\mathbf{x}^{\alpha+\beta}) = z_{\alpha+\beta}, \quad \forall \alpha, \beta \in \mathbb{N}_i^n.$$

Localizing matrix

Let q be the polynomial $\mathbf{x} \mapsto q(\mathbf{x}) := \sum_u q_u \mathbf{x}^u$. The localizing matrix $\mathbf{M}_i(q\,\mathbf{z})$ associated with $q \in \mathbb{R}[\mathbf{x}]$ and a sequence $\mathbf{z} = (z_\alpha)$, has its rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$, and with entries.

$$\mathbf{M}_i(q\,\mathbf{z})(\alpha, \beta) = L_{\mathbf{z}}(q(\mathbf{x})\mathbf{x}^{\alpha+\beta}) = \sum_{u \in \mathbb{N}^n} q_u z_{\alpha+\beta+u}, \quad \forall \alpha, \beta \in \mathbb{N}_i^n.$$

A sequence $\mathbf{z} = (z_\alpha) \subset \mathbb{R}$ is said to have a *representing* finite Borel measure supported on $\mathbf{K}$ if there exists a finite Borel measure μ such that

$$z_\alpha = \int_{\mathbf{K}} \mathbf{x}^\alpha \, d\mu, \quad \forall \alpha \in \mathbb{N}^n.$$

3.1 A "joint + marginal" approach

With $\{f, (g_j)_{j=1}^m\} \subset \mathbb{R}[\mathbf{x}]$, let $\mathbf{K} \subset \mathbb{R}^n$ be the basic semi-algebraic set

$$\mathbf{K} := \{\mathbf{x} \in \mathbb{R}^n \,:\, g_j(\mathbf{x}) \geq 0, \; j = 1, \ldots, m\}, \tag{3.1}$$

and consider the 0/1 polynomial optimization problem:

$$\mathbf{P}: \qquad f^* = \min\{f(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K} \cap \{0,1\}^n\}. \tag{3.2}$$

Let $\mathbf{Y} := \{0, 1\}$ and $0 < p < 1$, and with $y \in \mathbf{Y}$, define the parametrized 0/1 polynomial program in $n - 1$ variables

$$\mathbf{P}_y: \qquad J_1(y) = \min_{\mathbf{x}}\{f(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K} \cap \{0,1\}^n;\; x_1 = y\}, \tag{3.3}$$

or, equivalently $J_1(y) = \min\{f(\mathbf{x}) \,:\, \mathbf{x} \in \mathbf{K}_y\}$, where for every $y \in \mathbf{Y}$:

$$\mathbf{K}_y := \{\mathbf{x} \in \mathbf{K} \cap \{0,1\}^n \,:\, x_1 = y\}. \tag{3.4}$$

Observe that by definition, $f^* = \min_y \{J_1(y) \,:\, y \in \mathbf{Y}\} = \min[J_1(1), J_1(0)]$.

Semidefinite relaxations

To compute (or at least approximate) the optimal value f^* of problem $\mathbf{P}$ in (3.2), we now provide a hierarchy of SDP relaxations in the spirit of those defined in Lasserre [5]. Define the polynomials:

$$\mathbf{x} \mapsto u_k(\mathbf{x}) := x_k^2 - x_k, \quad k = 1, \ldots, n,$$

and let $v_j := \lceil(\deg g_j)/2\rceil$, $j = 1, \ldots, m$. For $k \geq \max_j v_j$ and some fixed $p \in (0, 1)$, consider the semidefinite program:

$$\begin{aligned} \rho_k = \inf_{\mathbf{z}} \, & L_{\mathbf{z}}(f) \\ \text{s.t.} \;\; & \mathbf{M}_k(\mathbf{z}) \succeq 0, \; \mathbf{M}_{k-v_j}(g_j\,\mathbf{z}) \succeq 0, \quad j = 1, \ldots, m \\ & \mathbf{M}_{k-1}(u_i\,\mathbf{z}) = 0, \; i = 1, \ldots, n; \quad L_{\mathbf{z}}(1) = 1, \; L_{\mathbf{z}}(x_1) = p. \end{aligned} \tag{3.5}$$

This semidefinite program (3.5) can be simplified as done in Lasserre [6] to work modulo the ideal $\langle (x_i^2 - x_i) \rangle$. Get rid of the constraints $\mathbf{M}_{k-1}(u_i\,\mathbf{z}) = 0$, $i = 1, \ldots, n$, and replace every moment variable z_α with z_β where $\beta_i = 1$ if $\alpha_i > 0$, and $\beta_i = 0$ if $\alpha_i = 0$. Next, the row and column of $\mathbf{M}_k(\mathbf{z})$ associated with the monomial $\mathbf{x}^\alpha$ is identical to that associated with $\mathbf{x}^\beta$ and can be deleted; same thing with the localizing matrices $\mathbf{M}_k(g_j\mathbf{z})$.

Letting $g_0 := 0$, the dual of (3.5) reads:

$$\begin{aligned} \rho_k^* = \sup_{\lambda, (\sigma_j), (\psi_i)} \; & \lambda_0 + p\lambda_1 \\ \text{s.t.} \;\; & f - (\lambda_0 + \lambda_1 x_1) = \sigma_0 + \sum_{j=1}^m \sigma_j\, g_j + \sum_{i=1}^n \psi_i\, u_i \\ & \sigma_j \in \Sigma[\mathbf{x}], \quad \psi_i \in \mathbb{R}[\mathbf{x}], \quad 0 \leq j \leq m; \quad 1 \leq i \leq n \\ & \deg \sigma_j g_j \leq 2k, \quad \deg \psi_i \leq 2k - 2, \quad 0 \leq j \leq m; \quad 1 \leq i \leq n. \end{aligned} \tag{3.6}$$

Equivalently, and denoting by $\mathbb{R}[x_1]_t$ the space of polynomials of degree at most t, the above dual may be rewritten as:

$$\begin{aligned} \rho_k^* = \sup_{q, (\sigma_j), (\psi_i)} \; & pq(1) + (1-p)q(0) \quad \left(= \int_{\mathbf{Y}} q\,d\varphi \right) \\ \text{s.t.} \quad & f - q = \sigma_0 + \sum_{j=1}^m \sigma_j\, g_j + \sum_{i=1}^n \psi_i\, u_i \\ & q \in \mathbb{R}[x_1]_1; \quad \sigma_j \in \Sigma[\mathbf{x}], \quad \psi_i \in \mathbb{R}[\mathbf{x}], \quad 0 \leq j \leq m; \quad 1 \leq i \leq n \\ & \deg \sigma_j g_j \leq 2k, \; \deg \psi_k \leq 2k - 2, \quad 0 \leq j \leq m; \quad 1 \leq i \leq n. \end{aligned} \tag{3.7}$$

Observe that with $I_1 := \langle x_1^2 - x_1 \rangle$ being the ideal of $\mathbb{R}[x_1]$ generated by the polynomial $x_1^2 - x_1$, one may also replace the polynomial q in (3.7) with $\tilde{q} := q + I_1$, i.e. an element of $\mathbb{R}[x_1]/I_1$. Indeed it is also admissible and $p\tilde{q}(1) + (1-p)\tilde{q}(0)$ is the same.

Theorem 3.1 *Let* $\mathbf{K}$ *be as* (3.1), $\mathbf{Y} = \{0, 1\}$, *and* $0 < p < 1$. *Assume that for every* $y \in \mathbf{Y}$ *the set* $\mathbf{K}_y$ *in* (3.4) *is nonempty, and consider the SDP relaxations* (3.7). *Then as* $k \to \infty$:

(a) $\rho_k^* \uparrow p\, J_1(1) + (1-p)\, J_1(0) \quad (= \int_{\mathbf{Y}} J_1 d\varphi)$. *In fact finite convergence takes place, i.e., there exists an integer* k^* *such that* $\rho_{k^*} = \int_{\mathbf{Y}} J_1 d\varphi$.

(b) *Let* $(q_k, (\sigma_j^k, \psi_i^k))$ *be a nearly optimal solution of* (3.7), *e.g. such that* $pq_k(1) + (1 - p)q_k(0) \geq \rho_k^* - 1/k$. *Then* $q_k(y) \leq J_1(y)$ *for* $y \in \mathbf{Y}$, *and*

$$\lim_{k \to \infty} q_k(y) \;=\; J_1(y), \quad y = 0, 1. \tag{3.8}$$

Moreover if one defines

$$\tilde{q}_0 := q_0, \quad y \mapsto \tilde{q}_k(y) \;:=\; \max\,[\,\tilde{q}_{k-1}(y), q_k(y)\,], \quad i = 1, 2, \ldots,$$

then $\tilde{q}_k(y) \uparrow J(y)$ for $y = 0, 1$, i.e. pointwise monotone nondecreasing convergence takes place.

Proof (a) The first part follows from Lasserre [8, Theorem 3.3] (adapted to the discrete case $\mathbf{Y} = \{0, 1\}$) whereas the second statement is a consequence of Lasserre [6, Theorem 3.2]. On the other hand, (b) follows from Lasserre [8, Theorem 3.5]. □

We also have the following sufficient condition which permits to detect whether finite convergence has occured.

Lemma 3.2 *Let* $\mathbf{z}$ *be an optimal solution of the SDP relaxation* (3.5) *and assume that*

$$\operatorname{rank} \mathbf{M}_k(\mathbf{z}) = \operatorname{rank} \mathbf{M}_{k-v}(\mathbf{z}), \tag{3.9}$$

where $v = \max_j v_j$. Then $\rho_k = \rho_k^* = \int_{\mathbf{Y}} J_1 d\varphi$.

Proof The proof is the same as that of Theorem 3.6 in Lasserre [6] and also follows from Theorem 4.1 in Lasserre [9] in a more general context. □

So when the sufficient condition (3.9) of Lemma 3.2 holds, $\mathbf{z}$ is the sequence of moments of a measure μ^* on $\mathbb{R}^n$, supported on $\left(\{0\} \times \mathbf{X}^*_{\{0\}}\right) \cup \left(\{1\} \times \mathbf{X}^*_{\{1\}}\right)$ (see the notation of Theorem 2.6). More precisely,

$$\mu^* = p\,\nu_1 + (1-p)\,\nu_2,$$

where ν_1 (resp. ν_2) is supported on finitely many optimal solutions of $\mathbf{P}$ given that $x_1 = 0$ (resp. given that $x_1 = 1$). One may extract such optimal solutions as indicated in Lasserre [9, §4.3] and Henrion et al. [4].

Remark 3.3 If the set $\mathbf{K}_y$ is empty for either $y = 0$ or $y = 1$ then $\rho_k = +\infty$ provided k is sufficiently large. Indeed otherwise one may show that $\rho_k = \int_{\mathbf{Y}} J_1(y)d\phi(y) = (1-p)J_1(0) + pJ_1(1)$ for all $k \geq k^*$ for some index k^*, in contradiction with $J_1(0) = +\infty$ or $J_1(1) = +\infty$. However, if $k < k^*$ one may have $\rho_k < +\infty$ while $J_1(0) = +\infty$ or $J_1(1) = +\infty$.

In what follows we will use the primal relaxation (3.5) with index k *fixed*. Hence since k will be fixed, it is important to check whether ρ_k is finite.

A sufficiency test for persistency

In the context of 0/1 deterministic optimization, *persistency* of a boolean variable x_i in a 0/1 program $\mathbf{P}$ is concerned with whether one may determine if either $x_i^* = 1$ or $x_i^* = 0$ in any optimal solution $\mathbf{x}^* \in \{0, 1\}^n$ of $\mathbf{P}$; see e.g. Bertsimas et al. [2], Natarajan et al. [13] and also the LP-based RLT relaxations described in Adams et al. [1].

There is a simple sufficient condition to detect whether $x_1 = 0$ (resp. $x_1 = 1$) cannot happen in any feasible solution of $\mathbf{P}$ defined in (3.2), in which case one may safely state that $x_1^* = 1$ (resp. $x_1^* = 0$) in *any* optimal solution $\mathbf{x}^* \in \{0, 1\}^n$.

Corollary 3.4 *Consider the SDP relaxation* (3.5) *with $\mathbf{x} \mapsto f(\mathbf{x}) := x_1$ (resp. $\mathbf{x} \mapsto f(\mathbf{x}) := -x_1$) and without the marginal constraint $L_{\mathbf{z}}(x_1) = p$. Denote its optimal value by ρ_k^0 (resp. ρ_k^1).*

If $\rho_k^0 > 0$ (resp. $-\rho_k^1 < 1$) then $x_1^* = 1$ (resp. $x_1^* = 0$) in any optimal solution $\mathbf{x}^* \in \{0, 1\}^n$ of $\mathbf{P}$.

Proof This is because ρ_k^0 (resp. ρ_k^1) always produces a lower bound on f^*. □

Hence the difficult case is when solving (3.5) with $f = x_1$ and $f = -x_1$, one obtains $\rho_k^0 = 0$ and $\rho_k^1 = -1$, respectively. But then the SDP relaxation (3.5) with f as in (3.2) is well-defined, that is, ρ_k is finite.

Corollary 3.5 *Let* $\mathbf{P}$ *be the 0/1 problem defined in* (3.2) *and let* ρ_i *be the optimal value associated with the SDP relaxation* (3.5). *If* $\rho_k^0 = 0$ *and* $\rho_k^1 = -1$ *(as defined in Corollary* 3.4*) then* ρ_k *is finite.*

Proof Let $\mathbf{z}_1$ (resp. $\mathbf{z}_2$) be an optimal solution of (3.5) associated with $f = x_1$ (resp. $f = -x_1$) without the marginal constraint $L_{\mathbf{z}}(x_1) = p$, and with optimal value $\rho_k^0 = 0$ (resp. $\rho_k^1 = -1$). Then the sequence $\mathbf{z} = p\,\mathbf{z}_2 + (1-p)\mathbf{z}_1$ is feasible for (3.5), hence with finite value. Indeed, $L_{\mathbf{z}_1}(x_1) = 0$ and $L_{\mathbf{z}_2}(x_1) = 1$ so that by linearity $L_{\mathbf{z}}(x_1) = L_{(1-p)\mathbf{z}_1+p\mathbf{z}_2}(x_1) = p$. □

3.2 The "joint + marginal" heuristic

Fix i and $0 < p < 1$. Denote by $\mathbf{x}_j \in \mathbb{R}^{n-j+1}$ the vector $(x_j, \ldots, x_n)$. For every $j = 2, \ldots, n$, and $\tilde{\mathbf{x}}_{j-1} = (\tilde{x}_1, \ldots, \tilde{x}_{j-1}) \in \{0,1\}^{j-1}$, let $\tilde{f}_j(\mathbf{x}_j) := f(\tilde{\mathbf{x}}_{j-1}, \mathbf{x}_j)$, and $\tilde{g}_\ell^j(\mathbf{x}_j) := g_\ell(\tilde{\mathbf{x}}_{j-1}, \mathbf{x}_j)$, $\ell = 1, \ldots, m$. Similarly, let

$$\mathbf{K}_j := \{\mathbf{x}_j \in \mathbb{R}^{n-j+1} : \tilde{g}_\ell^j(\mathbf{x}_j) \geq 0, \quad \ell = 1, \ldots, m\},$$

and given $\tilde{\mathbf{x}}_{j-1} \in \{0,1\}^{j-1}$, let $\mathbf{P}(\tilde{\mathbf{x}}_j)$ denote the parametric optimization problem

$$\mathbf{P}(\tilde{\mathbf{x}}_j): \quad \rho(\tilde{\mathbf{x}}_j) = \min_{\mathbf{x}_j} \{\tilde{f}_j(\mathbf{x}_j) : \mathbf{x}_j \in \mathbf{K}_j \cap \{0,1\}^{n-j+1};\ x_j = \tilde{x}_j\}, \tag{3.10}$$

with parameter $\tilde{x}_j$, i.e., the original problem $\mathbf{P}$ with the variables x_i, $i = 1, \ldots, j-1$, already fixed at the value $\tilde{x}_i \in \{0,1\}$, $i = 1, \ldots, j-1$, and where we now fix x_j at the value $\tilde{x}_j$.

With $\mathbf{z}$ being a sequence indexed in the monomial basis of $\mathbb{R}[\mathbf{x}_j]$, the associated SDP relaxation (3.5) reads:

$$\begin{aligned} \rho_{jk} = \inf_{\mathbf{z}}\ & L_{\mathbf{z}}(\tilde{f}_j) \\ \text{s.t.}\ & \mathbf{M}_k(\mathbf{z}) \succeq 0,\ \mathbf{M}_{k-v_\ell}(\tilde{g}_\ell^j\,\mathbf{z}) \succeq 0, \quad \ell = 1, \ldots, m \\ & \mathbf{M}_{k-1}(u_i\,\mathbf{z}) = 0, \quad i = j, \ldots, n; \quad L_{\mathbf{z}}(1) = 1, \quad L_{\mathbf{z}}(x_j) = p, \end{aligned} \tag{3.11}$$

with associated dual:

$$\begin{aligned} \rho_{jk}^* = \sup_{\lambda,(\sigma_i)}\ & \lambda_0 + p\lambda_1 \\ \text{s.t.}\ & \tilde{f}_j - (\lambda_0 + \lambda_1 x_j) = \sigma_0 + \sum_{\ell=1}^m \sigma_\ell\,\tilde{g}_\ell^j + \sum_{i=j}^n \psi_i\,u_i \\ & \sigma_\ell \in \Sigma[\mathbf{x}_j], \quad \psi_i \in \mathbb{R}[\mathbf{x}_j], \quad \ell = 0, \ldots, m;\ i = j, \ldots, n \\ & \deg\ \sigma_\ell \tilde{g}_\ell^j \leq 2k, \quad \deg\ \psi_i \leq 2k-2, \quad \ell = 0, \ldots, m;\ i = j, \ldots, n. \end{aligned} \tag{3.12}$$

The "joint + marginal" heuristic

For several problems like e.g., knapsack or k-cluster, whenever $\mathbf{P}(\tilde{\mathbf{x}}_j)$ in (3.10) is feasible, it has an obvious feasible solution denoted $\mathbf{s}_{j+1} \in \{0,1\}^{n-j}$.

For instance, the 0/1 knapsack problem $\mathbf{P}(\tilde{\mathbf{x}}_{j-1}) : \min_{\mathbf{x}_j} \{\mathbf{c}'_j \mathbf{x}_j : \sum_{\ell=j}^n a_\ell \, x_\ell \leq b(\tilde{\mathbf{x}}_{j-1})\}$, has the obvious feasible solution $\mathbf{s}_j = 0$ whenever $b(\tilde{\mathbf{x}}_{j-1}) \geq 0$, and no solution otherwise.

Similarly, the k-cluster problem $\mathbf{P}(\tilde{\mathbf{x}}_{j-1}) : \min_{\mathbf{x}_j} \{\mathbf{x}'_j Q \mathbf{x}_j : \sum_{\ell=j}^n x_\ell = k(\tilde{\mathbf{x}}_{j-1})\}$ (with $k(\tilde{\mathbf{x}}_{j-1}) \in \mathbb{Z}$), has no solution if $k(\tilde{\mathbf{x}}_{j-1}) < 0$ or $k(\tilde{\mathbf{x}}_{j-1}) > n-j+1$, has the obvious optimal solution $\mathbf{s}_j = 0$ if $k(\tilde{\mathbf{x}}_{j-1}) = 0$, and whenever $0 < \tilde{\mathbf{x}}_{j-1} \leq n - j$, it has the obvious feasible solution $\mathbf{s}_j = (s_{j\ell})$ with $s_{j\ell} = 1$, $\ell = 1, \ldots, k(\tilde{\mathbf{x}}_{j-1})$, and $s_{j\ell} = 0$, $\ell = k(\tilde{\mathbf{x}}_{j-1}) + 1, \ldots, n$.

The (J + M)-heuristic consists of n steps. At step j of the algorithm, the vector $\tilde{\mathbf{x}}_{j-1} = (\tilde{x}_1, \ldots, \tilde{x}_{j-1})$ (already computed) is such that $\mathbf{P}(\tilde{\mathbf{x}}_{j-1})$ has a feasible solution $\mathbf{s}_j \in \{0,1\}^{n-j+1}$. For the first step $j = 1$, one has: $\tilde{\mathbf{x}}_0 = \emptyset$, $f_j = f$ and $\tilde{g}_k^1 = g_k$, $k = 1, \ldots, m$:

Step j: **Input:** $\tilde{\mathbf{x}}_{j-1} \in \{0,1\}^{j-1}$. **Output:** $\tilde{\mathbf{x}}_j = (\tilde{\mathbf{x}}_{j-1}, \tilde{x}_j) \in \{0,1\}^j$.

Consider the SDP relaxation (3.11) for problem $\mathbf{P}(\tilde{\mathbf{x}}_j)$:

- Compute ρ_{jk}^0 for the SDP relaxation (3.11) without the moment constraint $L_\mathbf{z}(x_j) = p$ (see Corollary 3.4). If $\rho_{jk}^0 > 0$ then set $\tilde{x_j} = 1$ else compute ρ_{jk}^1; if $-\rho_{jk}^1 < 1$ then set $\tilde{x_j} = 0$.
- Else if $\rho_{jk}^0 = 0$ and $\rho_{jk}^1 = -1$ compute ρ_{jk} in (3.11) and extract λ_0, λ_1 from the dual (3.12). If $\lambda_1 < 0$ set $\tilde{x}_j = 1$, otherwise set $\tilde{x}_j = 0$.
- **Feasibility.**[1] If $\mathbf{P}((\tilde{\mathbf{x}}_{j-1}, \tilde{x}_j))$ has a feasible solution $\mathbf{s}_{j+1} \in \{0,1\}^{n-j}$ then set $\tilde{\mathbf{x}}_j = (\tilde{\mathbf{x}}_{j-1}, \tilde{x}_j)$, otherwise set $\tilde{\mathbf{x}}_j = (\tilde{\mathbf{x}}_{j-1}, 1 - \tilde{x}_j)$.

Repeat until $j = n$.

Computational complexity

At step j of the (J + M)-heuristic, one has to solve three semidefinite programs (3.11) (two of them without the moment constraint $L_\mathbf{z}(x_j) = p$) whose number of variables is $O((n-j+1)^k)$ and with m semidefinite constraints with matrix size at most $O((n-j)^k)$. Observe that when $k = 1$, the semidefinite program (3.11) has exactly same computational complexity as the standard kth SDP relaxation for 0/1 programs with $n - j + 1$ variables! Indeed the only difference is the single additional (linear) moment constraint $L_\mathbf{z}(x_j) = p$. For instance, for the MAXCUT problem:

$$\max_{\mathbf{x}} \left\{ \frac{1}{2} \sum_{i \leq i < j \leq n} Q_{ij}(1 - x_i x_j) \; : \; \mathbf{x} \in \{-1, 1\}^n \right\},$$

where $Q = Q' = (Q_{ij}) \in \mathbb{R}^{n \times n}$ with $Q_{ii} = 0$ for all i, and with $k = 1$, the SDP relaxation (3.11) at step 1 of the (J + M)-heuristic reads:

$$\max_{\mathbf{x}, \mathbf{X}} \left\{ -\frac{1}{2} \text{trace}\,(Q\,\mathbf{X}) \; : \; \mathbf{X}_{ii} = 1; \quad \begin{bmatrix} 1 & \mathbf{x}' \\ \mathbf{x} & \mathbf{X} \end{bmatrix} \succeq 0, \right. \\ \left. \mathbf{X}' = \mathbf{X} \in \mathbb{R}^{n \times n}; \quad x_1 = 1 - 2p \right\}, \tag{3.13}$$

which is the standard Goemans and Williamson (or Shor) SDP relaxation with the single additional constraint $x_1 = 1 - 2p$. At step $j > 1$, and with $n_j := n - j + 1$, (3.11) reads:

[1] When k is large enough, feasiblity is guaranteed because $+\infty > \rho_{jk} \approx \mathbf{P}(\tilde{\mathbf{x}}_j)$ implies that there is a solution $\mathbf{x}^* \in \mathbf{K} \cap \{0,1\}^n$ with $\mathbf{x}_\ell^* = \tilde{x}_\ell$, $\ell = 1, \ldots, j$. So the above feasibility test is conservative, in case k is not large enough.

$$\max_{\mathbf{x},\mathbf{X}} \left\{ \mathbf{c}_j'\mathbf{x}_j + \mathrm{trace}\,(Q_j\,\mathbf{X}) : \mathbf{X}_{ii} = 1;\ \begin{bmatrix} 1 & \mathbf{x}_j' \\ \mathbf{x}_j & \mathbf{X} \end{bmatrix} \succeq 0, \right. \tag{3.14}$$

$$\left. \mathbf{X}' = \mathbf{X} \in \mathbb{R}^{n_j \times n_j};\ x_j = 1 - 2p \right\},$$

for some appropriate vector $\mathbf{c}_j \in \mathbb{R}^{n-j+1}$, and matrix $Q_j \in \mathbb{R}^{n_j \times n_j}$.

3.3 Variants

We here briefly describe two variants of the basic (J + M)-heuristic of Sect. 3.2.

3.3.1 *The max-gap variant*

In what we call the "max-gap" variant of the (J + M)-heuristic, at each step j one may try to optimize the choice of the variable to treat as parameter instead of the simple choice x_1 at $j = 1$, then x_2 at $j = 2$, etc. For instance, at step $j = 1$, solve the SDP relaxation (3.11) with x_i as parameter, and get $(\lambda_0^i, \lambda_1^i)$ from an optimal solution of the dual, $i = 1, \ldots, n$. Then select the index i for which $|\lambda_1^i|$ is maximum, and fix $\tilde{x}_i = 1$ if $\lambda_1^i < 0$ and $\tilde{x}_i = 0$ otherwise. The rationale behind this variant is that the larger $|\lambda_1^i|$ is, the better is the approximation $|q_k(0) - q_k(1)|$ of $|J_i(0) - J_i(1)|$, and so the more likely the decision $x_i = 0$ or $x_i = 1$ is correct. Then repeat in the obvious manner with now the remaining variables $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, etc. Of course this variant is computationally more demanding since one must solve nkth SDP relaxations instead of one, then $n - 1, n - 2$, etc.

3.3.2 *Compound variant*

As already mentioned in introduction, one may also easily consider groupings of, say $s > 1$, variables at a time, instead of one variable at a time in the basic (J + M)-heuristic. Recall that $\mathbb{N}_k^s = \{\alpha \in \mathbb{N}^s : \sum_i \alpha_i \leq k\}$. Let $\mathbf{Y} = \{0, 1\}^s$ and let φ be the probability uniformly supported on $\mathbf{Y}$, with moments

$$\gamma_\beta = \int_{\mathbf{Y}} \mathbf{x}^\beta \, d\varphi = 2^{-s} \sum_{\mathbf{x} \in \mathbf{Y}^s} \mathbf{x}^\beta, \quad \forall \beta \in \mathbb{N}_{2k}^s.$$

Let $\Gamma_k^s := \{\alpha \in \mathbb{N}_{2k}^s : \sum_{i=1}^s \alpha_i \leq 2k;\ \alpha_i \leq 1, \forall i\}$. The first kth SDP relaxation now reads

$$\rho_k = \inf_{\mathbf{z}} L_{\mathbf{z}}(f) \tag{3.15}$$

$$\text{s.t.} \quad \mathbf{M}_k(\mathbf{z}) \succeq 0,\ \mathbf{M}_{k-v_j}(g_j\,\mathbf{z}) \succeq 0, \quad j = 1, \ldots, m$$

$$\mathbf{M}_{k-1}(u_i\,\mathbf{z}) = 0,\ i = 1, \ldots, n; \quad L_{\mathbf{z}}(1) = 1,\ L_{\mathbf{z}}(\mathbf{x}^\beta) = \gamma_\beta, \quad \forall\,\beta \in \Gamma_k^s,$$

that is, the moment constraints $L_z(\mathbf{x}^\beta) = \gamma_\beta$ are only concerned with the monomials $\mathbf{x}^\beta = \mathbf{x}_1^{\beta_1} \cdots x_s^{\beta_s}$ of degree at most $2k$, and with $\beta_i \leq 1$ for all $i = 1, \ldots, s$. Observe that

$$|\Gamma_k^s| = \begin{cases} 2^s & \text{if } 2k \geq s \\ \displaystyle\sum_{\ell=0}^{2k} \binom{s}{\ell} & \text{if } 2k < s. \end{cases}$$

The dual reads

$$\begin{aligned}\rho_k^* = \sup_{\lambda,(\sigma_j),(\psi_i)} & \sum_{\beta\in\Gamma_k^s} \gamma_\beta\lambda_\beta \\ \text{s.t. } & f - \sum_{\beta\in\Gamma_k^s}\lambda_\beta \mathbf{x}^\beta = \sigma_0 + \sum_{j=1}^m \sigma_j\, g_j + \sum_{i=1}^n \psi_i\, u_i \\ & \sigma_j \in \Sigma[\mathbf{x}], \quad \psi_i \in \mathbb{R}[\mathbf{x}], \quad 0 \le j \le m;\ 1 \le i \le n \\ & \deg \sigma_j g_j \le 2k, \quad \deg \psi_k \le 2k-2, \quad 0 \le j \le m;\ 1 \le i \le n.\end{aligned} \tag{3.16}$$

Let $q \in \mathbb{R}[x_1, \ldots, x_s]_{2k}$ be the polynomial

$$\mathbf{x} \mapsto q(\mathbf{x}) := \sum_{\beta\in\Gamma_k^s} \lambda_\beta^* \mathbf{x}^\beta,$$

of degree at most $2k$, obtained from an optimal solution (λ^*) of (3.16).

One computes $\mathbf{u}_s = (u_1, \ldots, u_s) \in \mathbf{Y}^s$ which minimizes q on $\mathbf{Y}^s$ by inspection of the 2^s values of q on $\mathbf{Y}$. For feasibility, one proceeds one variable by one. If $\mathbf{P}(u_1)$ has a feasible solution then one sets $\tilde{x}_1 = u_1$ and $\tilde{x}_1 = 1-u_1$ otherwise. If $\mathbf{P}(\tilde{x}_1, u_2)$ has a feasible solution then one sets $\tilde{x}_2 = u_2$ and $\tilde{x}_2 = 1 - u_2$ otherwise, etc. until one obtains $\tilde{\mathbf{x}}_s \in \mathbf{Y}^s$ for which $\mathbf{P}(\tilde{\mathbf{x}}_s)$ has a feasible solution. Then one iterates with problem $\mathbf{P}$ in which $(x_1, \ldots, x_s)$ is fixed, equal to $\tilde{\mathbf{x}}_s$, etc. in the obvious way.

4 Computational experiments

We report on a first set of computational experiments on the MAXCUT, k-cluster and 0/1 knapsack problems. All experiments were run on a Intel(R) Core(TM)2 Due CPU 2.00 GHz with 2 Gb ram.

4.1 The MAXCUT problem

The celebrated MAXCUT problem formally consists of solving the optimization problem:

$$\mathbf{P}: \quad \max_{\mathbf{x}} \left\{ \frac{1}{2} \sum_{1\le i<j\le n} Q_{ij}\,(1 - x_i x_j) : \mathbf{x} \in \{-1, 1\}^n \right\},$$

for some real symmetric matrix $Q = (Q_{ij}) \in \mathbb{R}^{n\times n}$.

In our sample of randomly generated problems, the entry Q_{ij} of the real symmetric matrix Q is set to zero with probability 1/2 and when different from zero, Q_{ij} is randomly (and independently) generated according to the uniform probability distribution on the interval [0, 10].

We have tested the basic version of the (J + M)-heuristic (see Sect. 3.3.1) with $k = 1$ and $p = 1/2$, i.e. solving the SDP-relaxations (3.13)–(3.14), for MAXCUT problems on random graphs with $n = 20, 30, 40$ and 50 variables. For each value of n, we have generated 50 problems (and 20 for $n = 50$). In (3.5) the parameter $p \in (0, 1)$ is set to 0.5. Let $\mathbf{Q}_1$ denote the optimal value of the SDP-relaxation (3.13) without the marginal constraint $x_1 = p$, that is, $\mathbf{Q}_1$ is the Shor's relaxation with famous Goemans and Williamson's 0.878 performance guarantee (for maximization and nonnegative weights Q_{ij}). Let $\mathbf{P}_1$ denote the cost of the

Table 1 Relative error for MAXCUT; $p = 0.5$

n	20 (%)	30 (%)	40 (%)	50 (%)
$(\mathbf{Q}_1 - \mathbf{P}_1)/\vert\mathbf{Q}_1\vert$	3.23	3.28	3.13	2.92
$(\mathbf{Q}_1 - \mathrm{GW})/\vert\mathbf{Q}_1\vert$	2.58	2.60	2.84	2.60

solution $\mathbf{x} \in \{-1, 1\}^n$ generated by the (J + M)-heuristic.[2] In Table 1, we have reported the average relative error $(\mathbf{Q}_1 - \mathbf{P}_1)/|\mathbf{Q}_1|$, which, as one may see, is very small and comparable with the relative error $(\mathbf{Q}_1 - \mathrm{GW})/|\mathbf{Q}_1|$ obtained from the Goemans and Williamson (GW) solution. The latter was obtained by the randomized rounding procedure described in Goemans and Williamson [3] with a sample size of 50 as recommended in Goemans and Williamson [3] for $n \approx 50$ and we used the `schur` MATLAB subroutine to compute eigenvectors. The value of the GW-solution is significantly better than the theoretical bound, in accordance with the computational results displayed in Goemans and Williamson [3]. For $n = 50$ the (J + M)-solution was better than the GW-solution in 7 out of the 20 randomly generated problems. For $n = 40$ (resp. $n = 50$) variables the CPU time was 218 s (resp. 945 s) whereas it was 232 s (resp. 507 s) for the GW solution. The reader should keep in mind that the (J + M)-heuristic is not specific to the MAXCUT problem and was run with the smallest possible choice $k = 1$ of the parameter k.

4.2 The k-cluster problem

We have also tested the (J + M)-heuristic for the k-cluster problem:

$$\mathbf{P}: \quad \max_{\mathbf{x}} \left\{ \mathbf{x}' Q \mathbf{x} \;:\; \mathbf{x} \in \{0, 1\}^n;\; \sum_{\ell=1}^{n} x_\ell = k \right\}, \tag{4.1}$$

again for some real symmetric matrix $Q = (Q_{ij}) \in \mathbb{R}^{n \times n}$, and some fixed integer $k \in \mathbb{N}$, $1 \leq k < n$. Observe that the constraint $\sum_\ell x_\ell = k$ is linear. Therefore, to take full advantage of the kth SDP relaxation (3.11) which contains moments z_α of order up to $2k$, one may add the n constraints $x_i(k - \sum_\ell x_\ell) = 0$, $i = 1, \ldots, n$, in the definition (4.1) of $\mathbf{P}$ because they are redundant. However these constraints make the kth SDP relaxation more constrained. They also correspond to the first-level RLT constraints defined in Sherali and Adams [14].

As for MAXCUT, $\mathbf{Q}_1$ denotes the optimal value of the first SDP-relaxation in the hierarchy to solve problem $\mathbf{P}$, i.e.,

$$\mathbf{Q}_1: \quad \begin{cases} \max\limits_{\mathbf{x},\mathbf{X}} \; \mathrm{trace}\,(Q\,\mathbf{X}) \\ \text{s.t.} \quad \begin{bmatrix} 1 & \mathbf{x}' \\ \mathbf{x} & \mathbf{X} \end{bmatrix} \succeq 0, \quad \mathbf{X}' = \mathbf{X} \in \mathbb{R}^{n \times n} \\ \qquad X_{ii} = x_i, \quad i = 1, \ldots, n \\ \qquad \sum_{\ell=1}^{n} x_i = k \\ \qquad k\,x_i - \sum_{\ell=1}^{n} X_{i\ell} = 0, \quad i = 1, \ldots, n \end{cases}$$

whereas $\mathbf{P}_1$ denote the cost of the solution $\mathbf{x} \in \{0, 1\}^n$ generated by the (J + M)-heuristic.

[2] $\mathbf{Q}_1$ and $\mathbf{P}_1$ were computed with the GloptiPoly software dedicated to solving the generalized problem of moments [4].

We have tested the (J + M)-heuristic on problems randomly generated as for MAXCUT, and with $k = n/2 = 10$. The average relative error $\mathbf{Q}_1 - \mathbf{P}_1|/|\mathbf{Q}_1|$ was

- 5.7% on 4 randomly generated problems with $n = 60$ variables,
- 4.5 and 5.6% on 2 randomly generated problems with $n = 70$ variables. The "max-gap" variant was a little better (≈ 4 and ≈ 4.5% respectively).
- 5.7% on a problem with $n = 80$ variables.

The CPU times were of the same order of magnitude as for the MAXCUT problem.

4.3 The 0/1 knapsack problem

Finally, we have also tested the (J + M)-heuristic for the 0/1 knapsack problem:

$$\mathbf{P}: \quad \max_{\mathbf{x}} \left\{ \mathbf{c}'\mathbf{x} \,:\, \mathbf{x} \in \{0,1\}^n;\ \sum_{\ell=1}^{n} a_\ell\, x_\ell \,\leq\, b \right\}, \tag{4.2}$$

for some real vector $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{a} \in \mathbb{N}^n$, $b \in \mathbb{N}$.

As for the k-cluster problem, the constraint $\sum_\ell a_\ell\, x_\ell \leq b$ is linear. Therefore, again, to take full advantage of the kth SDP relaxation (3.11) which contains moments z_α of order up to $2k$, one may add the n redundant constraints $x_i\,(b - \sum_\ell a_\ell x_\ell) \geq 0$, and $(1 - x_i)(b - \sum_\ell a_\ell x_\ell) \geq 0$, $i = 1, \ldots, n$, in the definition (4.2) of $\mathbf{P}$; they also correspond to the first-level RLT constraints in Sherali and Adams [14].

Again, and as for MAXCUT and k-cluster problems, $\mathbf{Q}_1$ denotes the optimal value of the first SDP-relaxation in the hierarchy to solve problem $\mathbf{P}$, i.e.,

$$\mathbf{Q}_1: \quad \left\{ \begin{array}{ll} \displaystyle\max_{\mathbf{x},\mathbf{X}} \ \mathbf{c}'\mathbf{x} & \\ \text{s.t.} & \begin{bmatrix} 1 & \mathbf{x}' \\ \mathbf{x} & \mathbf{X} \end{bmatrix} \succeq 0, \quad \mathbf{X}' = \mathbf{X} \in \mathbb{R}^{n\times n} \\ & \\ & X_{ii} = x_i, \ i = 1, \ldots, n \\ & b\,x_i - \displaystyle\sum_{\ell=1}^{n} a_\ell X_{i\ell} \geq 0, \quad i = 1, \ldots, n \\ & b - b\,x_i - \displaystyle\sum_{\ell=1}^{n} a_\ell (x_\ell - X_{i\ell}) \geq 0, \quad i = 1, \ldots, n \end{array} \right.$$

whereas $\mathbf{P}_1$ denote the cost of the solution $\mathbf{x} \in \{0,1\}^n$ generated by the (J + M)-heuristic.

We have tested the "joint + marginal" heuristic on a sample of 16 problems with $n = 50$ variables and 3 problems with $n = 60$ variables where, $b = \sum_\ell a_\ell/2$, and the integers a_ℓ's are generated uniformly in [10, 100]. The vector $\mathbf{c}$ is generated by: $c_\ell = s\epsilon + a_\ell$ with $s = 0.1$ and ϵ is a random variable uniformly distributed in [0, 1]. From the results reported in Table 2, one may see that very good relative errors are obtained, in accordance with the fact that the 0/1 knapsack problem is considered as an easy problem since there is a Fully Polynomial Time Approximation Scheme. Moreover, recently, Karlin et al. [12] have shown

Table 2 Relative error for 0/1 knapsack

n	50 (%)	60 (%)
$(\mathbf{Q}_1 - \mathbf{P}_1)/\lvert\mathbf{Q}_1\rvert$	2.1	0.62

that solving the tth SDP-relaxation of the Lasserre hierarchy produces an integrality gap of at most $t/(t-1)$. The CPU time is very comparable (for same value of n) to the one for MAXCUT and k-cluster problems since we solve the same sequence of very similar semidefinite programs.

Discussion

Concerning sensitivity to the parameter $p \in (0, 1)$, we have also tried the values $p = 1/3$, and $p = 3/4$ on the MAXCUT problem without any significant change in the quality of the results. For instance, on a sample of 50 randomly generated examples with $n = 40$ variables, the average relative error was 4.5% for $p = 0.25$ and 3.79% for $p = 0.75$ (to be compared with 3.13% for $p = 0.5$). Moreover, the running time is independent of p (taking for granted that two semidefinite programs whose data differ at the right-hand-side of a single equality constraint have same running time).

Also, we have tried the compound variant with two variables, i.e., after solving the first SDP-relaxation we test four values of the polynomial $J_1(\tilde{x}_1, \tilde{x}_2)$ [at the points $(0, 0)$, $(1, 0)$, $(0, 1)$, $(1, 1)$] and branch on the point $(\tilde{x}_1, \tilde{x}_2)$ with smallest value. We then repeat the process on the $(n-2)$-variables problem $\mathbf{P}$ where (x_1, x_2) is fixed at the value $(\tilde{x}_1, \tilde{x}_2)$, etc. As mentioned, instead of the single linear constraint $L_{\mathbf{z}}(x_1) = p$, the first SDP-relaxation (3.11) has now three linear constraints $L_{\mathbf{z}}(x_1) = L_{\mathbf{z}}(x_2) = p$ and $L_{\mathbf{z}}(x_1x_2) = p^2$ with associated dual variables $\lambda_{10}, \lambda_{01}, \lambda_{11}$ in (3.12). Hence, the polynomial J_1 obtained from an optimal solution of the dual (3.12) is of the form $\lambda_0 + \lambda_{10}x_1 + \lambda_{01}x_2 + \lambda_{11}x_1x_2$. Tested on the MAXCUT problem with $n = 40$ nodes, this compound variant did not bring any improvement. The relative error was 7.14% for $p = 0.5$ and 5.55% for $p = 0.25$. A higher value of the parameter k (e.g. $k = 2$) would provide better performance but at the cost of a prohibitive CPU time.

5 Conclusion

We have proposed an algorithm for 0/1 programs which builds upon results from the "joint + marginal" approach developed in Lasserre [8]. The preliminary results presented here are very encouraging. Indeed, we have obtained good results in testing the (J + M)-heuristic with $k = 1$, that is, using moments of order up to 2 only (whereas the rationale behind this algorithm invokes potentially large k)! On the other hand, there is also a rationale behind the idea of using small values of k (typically 1 or 2) as it is well-known that for some combinatorial problems (like e.g. MAXCUT) the first SDP-relaxation (i.e. with $k = 1$) already provides good lower bounds (sometimes even with some guarantee). Nevertheless, there are several issues that remain to be investigated. For instance, it would also be interesting to test the (J + M)-heuristic that uses LP-relaxations with higher order, say for instance $k = 2$ (instead of the too costly SDP-relaxations (for $k = 2$), and compare the results with the one presented here with SDP-relaxations (and $k = 1$). Also the compound variant of Sect. 3.3.2 that uses marginals on 2, 3, or say s, variables (with s relatively small anyway) instead of 1 variable in the present version, and using again LP-relaxations with higher order $k = 2$. Finally, how to adapt the method for 0/1 problems where the feasibility issue is crucial?

Acknowledgments The authors are grateful to anonymous referees for very helpful remarks and suggestions that helped preparing this revised version.

References

1. Adams, W.P., Bowers Lassiter, J., Sherali, H.D.: Persistency in 0-1 polynomial programming. Math. Oper. Res. **23**, 359–389 (1998)
2. Bertsimas, D., Natarajan, K., Teo, C.-P.: Persistence in discrete optimization under data uncertainty. Math. Program. Ser. B **108**, 251–274 (2005)
3. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. Assoc. Comput. Mach **42**, 1115–1145 (1995)
4. Henrion, D., Lasserre, J.B., Lofberg, J.: GloptiPoly 3: moments, optimization and semidefinite programming. Optim. Methods Softw. **24**, 761–779 (2009)
5. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**, 796–817 (2001)
6. Lasserre, J.B.: An explicit equivalent positive semidefinite program for nonlinear 0/1 programs. SIAM J. Optim. **12**, 756–769 (2002)
7. Lasserre, J.B.: Semidefinite programming vs. LP relaxations for polynomial programming. Math. Oper. Res. **27**, 347–360 (2002)
8. Lasserre, J.B.: A "joint + marginal" approach to parametric polynomial optimization. SIAM J. Optim. **20**, 1995–2022 (2010)
9. Lasserre, J.B.: Moments, Positive Polynomials and Their Applications. Imperial College Press, London (2009)
10. Laurent, M.: A comparison of the Sherali–Adams, Lovász–Schrijver and Lasserre relaxations for 0/1 programming. Math. Oper. Res. **28**, 470–496 (2003)
11. Lovász, L., Schrijver, A.: Cones of matrices and set-functions for 0-1 optimization problems. SIAM J. Optim. **1**, 166–190 (1991)
12. Karlin, A.R., Mathieu, C., Thach Nguyen, C.: Integrality gaps of linear and semi-definite programming relaxations for knapsack. Available at `arXiv1007.1283v1`
13. Natarajan, K., Song, M., Teo, C.-P.: Persistency and its applications in choice modelling. Manag. Sci. **55**, 453–469 (2009)
14. Sherali, H., Adams, W.P.: A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM J. Discret Math. **3**, 411–430 (1990)

Résumé de la thèse:

L''Optimisation Polynomiale' s'intéresse aux problèmes d'optimisation P de la forme

min {f(x): x dans K} où f est un polynôme et K est un ensemble semi-algébrique de base, c'est-à-dire défini par un nombre fini de contraintes inégalité polynomiales, K={x dans Rn : gj(x) <= 0}. Cette sous discipline de l'optimisation a émergé dans la dernière décennie grâce à la combinaison de deux facteurs: l'existence de certains résultats puissants de géométrie algébrique réelle et la puissance de l'optimisation semidéfinie (qui permet d'exploiter les premiers). Il en a résulté une méthodologie générale (que nous appelons ``moments-SOS') qui permet d'approcher aussi près que l'on veut l'optimum global de P en résolvant une hiérarchie de relaxations convexes. Cependant, chaque relaxation étant un programme semi-défini dont la taille augmente avec le rang dans la hiérarchie, malheureusement, au vu de l'état de l'art actuel des progiciels de programmation semidéfinie, cette méthodologie est pour l'instant limitée à des problèmes P de taille modeste sauf si des symétries ou de la parcimonie sont présentes dans la définition de P. Cette thèse essaie donc de répondre à la question: Peux-t-on quand même utiliser la méthodologie moments-SOS pour aider à résoudre P même si on ne peut résoudre que quelques (voire une seule) relaxations de la hiérarchie? Et si oui, comment? Nous apportons deux contributions:

I. Dans une première contribution nous considérons les problèmes non convexes en variables mixtes (MINLP) pour lesquelles dans les contraintes polynomiales {g(x) <=0} où le polynôme g n'est pas concave, g est concerné par peu de variables. Pour résoudre de tels problèmes (de taille est relativement importante) on utilise en général des méthodes de type ``Branch-and-Bound'. En particulier, pour des raisons d'efficacité évidentes, à chaque nœud de l'arbre de recherche on doit calculer rapidement une borne inférieure sur l'optimum global. Pour ce faire on utilise des relaxations convexes du problème obtenues grâce à l'utilisation de sous estimateurs convexes du critère f (et des polynômes g pour les contraintes g(x)<= 0 non convexes). Notre contribution est de fournir une méthodologie générale d'obtention de tels sous estimateurs polynomiaux convexes pour tout polynôme g, sur une boite. La nouveauté de notre contribution (grâce à la méthodologie moment-SOS) est de pouvoir minimiser directement le critère d'erreur naturel qui mesure la norme L_1 de la différence f-f' entre f et son sous estimateur convexe polynomial f'. Les résultats expérimentaux confirment que le sous estimateur convexe polynomial que nous obtenons est nettement meilleur que ceux obtenus par des méthodes classiques de type ``alpha-BB' et leurs variantes, tant du point de vue du critère L_1 que du point de vue de la qualité des bornes inférieures obtenus quand on minimise f' (au lieu de f) sur la boite.

II: Dans une deuxième contribution on considère des problèmes P pour lesquels seules quelques relaxations de la hiérarchie moments-SOS peuvent être implantées, par exemple celle de rang k dans la hiérarchie, et on utilise la solution de cette relaxation pour construire une solution admissible de P. Cette idée a déjà été exploitée pour certains problèmes combinatoire en variables 0/1, parfois avec des garanties de performance remarquables (par exemple pour le problème MAXCUT). Nous utilisons des résultats récents de l'approche moment-SOS en programmation polynomiale paramétrique pour définir un algorithme qui calcule une solution admissible pour P à partir d'une modification mineure de la relaxation convexe d'ordre k. L'idée de base est de considérer la variable x_1 comme un paramètre dans un intervalle Y_1 de R et on approxime la fonction ``valeur optimale' J(y) du problème d'optimisation paramétrique P(y)= min {f(x): x dans K; x_1=y} par un polynôme univarié de degré d fixé. Cette étape se ramène à la résolution d'un problème d'optimisation convexe (programme semidéfini). On calcule un minimiseur global y de J sur l'intervalle Y (un problème d'optimisation convexe ``facile') et on fixe la variable x_1=y. On itère ensuite sur les variables restantes x_2,…,x_n en prenant x_2 comme paramètre dans un intervalle Y_2, etc. jusqu'à obtenir une solution complète x de R^n qui est faisable si K est convexe ou dans certains problèmes en variables 0/1 où la faisabilité est facile à vérifier (e.g., MAXCUT, k-CLUSTTER, Knapsack). Sinon on utilise le point obtenu x comme initialisation dans un procédure d'optimisation locale pour obtenir une solution admissible. Les résultats expérimentaux obtenus sur de nombreux exemples sont très encourageants et prometteurs.

Summary of the thesis:

Polynomial Optimization is concerned with optimization problems of the form (P) : f* = { f(x) with x in set K}, where K is a basic semi-algebraic set in Rn defined by K={x in Rn such as gj(x) less or equal 0}; and f is a real polynomial of n variables x = (x1, x2, ..., xn).

In this thesis we are interested in problems (P) where symmetries and/or structured sparsity are not easy to detect or to exploit, and where only a few (or even no) semidefinite relaxations of the moment-SOS approach can be implemented. And the issue we investigate is: How can the moment-SOS methodology be still used to help solve such problem (P)? We provide two applications of the moment-SOS approach to help solve (P) in two different contexts.

* In a first contribution we consider MINLP problems on a box B = [xL, xU] of Rn and propose a moment-SOS approach to construct polynomial convex underestimators for the objective function f (if non convex) and for -gj if in the constraint gj(x) less or equal 0, the polynomial gj is not concave. We work in the context where one wishes to find a convex underestimator of a non-convex polynomial f of a few variables on a box B of Rn. The novelty with previous works on this topic is that we want to compute a polynomial convex underestimator p of f that minimizes the important tightness criterion which is the L1 norm of (f-h) on B, over all convex polynomials h of degree d _fixed. Indeed in previous works for computing a convex underestimator L of f, this tightness criterion is not taken into account directly. It turns out that the moment-SOS approach is well suited to compute a polynomial convex underestimator p that minimizes the tightness criterion and numerical experiments on a sample of non-trivial examples show that p outperforms L not only with respect to the tightness score but also in terms of the resulting lower bounds obtained by minimizing respectively p and L on B. Similar improvements also occur when we use the moment-SOS underestimator instead of the aBB-one in refinements of the aBB method.

* In a second contribution we propose an algorithm that also uses an optimal solution of a semidefinite relaxation in the moment-SOS hierarchy (in fact a slight modification) to provide a feasible solution for the initial optimization problem but with no rounding procedure. In the present context, we treat the first variable x1 of x = (x1, x2, ..., xn) as a parameter in some bounded interval Y of R. Notice that f*=min { J(y) : y in Y} where J is the function J(y) := inf {f(x) : x in K ; x1=y}. That is one has reduced the original n-dimensional optimization problem (P) to an equivalent one-dimensional optimization problem on an interval. But of course determining the optimal value function J is even more complicated than (P) as one has to determine a function (instead of a point in Rn), an infinite-dimensional problem. But the idea is to approximate J(y) on Y by a univariate polynomial p(y) with the degree d and fortunately, computing such a univariate polynomial is possible via solving a semidefinite relaxation associated with the parameter optimization problem. The degree d of p(y) is related to the size of this semidefinite relaxation. The higher the degree d is, the better is the approximation of J(y) by p(y) and in fact, one may show that p(y) converges to J(y) in a strong sense on Y as d increases. But of course the resulting semidefinite relaxation becomes harder (or impossible) to solve as d increases and so in practice d is fixed to a small value. Once the univariate polynomial p(y) has been determined, one computes x1* in Y that minimizes p(y) on Y, a convex optimization problem that can be solved efficiently. The process is iterated to compute x2 in a similar manner, and so on, until a point x in Rn has been computed. Finally, as x* is not feasible in general, we then use x* as a starting point for a local optimization procedure to find a final feasible point x in K. When K is convex, the following variant is implemented. After having computed x1* as indicated, x2* is computed with x1 fixed at the value x1*, and x3 is computed with x1 and x2 fixed at the values x1* and x2* respectively, etc., so that the resulting point x* is feasible, i.e., x* in K. The same variant applies for 0/1 programs for which feasibility is easy to detect like e.g., for MAXCUT, k-CLUSTER or 0/1-KNAPSACK problems.