Minimizing the sum of many rational functions

Florian Bugarin, Didier Henrion, Jean-Bernard Lasserre

Minimizing the sum of many rational functions*

Florian Bugarin,[1] Didier Henrion,[2,3] Jean-Bernard Lasserre[2,4]

December 4, 2014

Abstract

We consider the problem of globally minimizing the sum of many rational functions over a given compact semialgebraic set. The number of terms can be large (10 to 100), the degree of each term should be small (up to 10), and the number of variables can be relatively large (10 to 100) provided some kind of sparsity is present. We describe a formulation of the rational optimization problem as a generalized moment problem and its hierarchy of convex semidefinite relaxations. Under some conditions we prove that the sequence of optimal values converges to the globally optimal value. We show how public-domain software can be used to model and solve such problems. Finally, we also compare with the epigraph approach and the BARON software.

Keywords: rational optimization; global optimization; semidefinite relaxations; sparsity.
AMS MSC 2010: 46N10, 65K05, 90C22, 90C26.

1 Introduction

Consider the optimization problem

$$f^* := \inf_{\mathbf{x}\in\mathbf{K}} \sum_{i=1}^{N} f_i(\mathbf{x}) \tag{1}$$

over the basic semi-algebraic set

$$\mathbf{K} := \{\mathbf{x} \in \mathbb{R}^n \,:\, g_j(\mathbf{x}) \geq 0,\ j = 1, \ldots, m\}, \tag{2}$$

[1]Institut Clément Ader; Modelling of Mechanical Systems and Microsystems Group; Université Paul Sabatier; Espace Clément Ader, 3 rue Caroline Aigle, F-31400 Toulouse.

[2]CNRS; LAAS; 7 avenue du colonel Roche, F-31400 Toulouse,France; Université de Toulouse, LAAS,F-31400 Toulouse; France.

[3]Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, CZ-16607 Prague, Czech Republic

[4]Institut de Mathématiques de Toulouse, Université de Toulouse, France

*The first author acknowledges support by the French National Research Agency (ANR) through COSINUS program (project ID4CS ANR-09-COSI-005). The second author acknowledges support by Research Program MSM6840770038 of the Czech Ministry of Education and Project 103/10/0628 of the Grant Agency of the Czech Republic.

for given polynomials $g_j \in \mathbb{R}[\mathbf{x}]$, $j = 1, \ldots, m$, and where each term $f_i : \mathbb{R}^n \to \mathbb{R}$ is a rational function

$$\mathbf{x} \mapsto f_i(\mathbf{x}) := \frac{p_i(\mathbf{x})}{q_i(\mathbf{x})},$$

with $p_i, q_i \in \mathbb{R}[\mathbf{x}]$ for each $i = 1, \ldots, N$.

Problem (1) is a *fractional programming* problem of a rather general form. Nevertheless, we assume that the degree of each f_i and g_j is relatively small (up to 10), but the number of terms N can be quite large (10 to 100). For dense data, the number of variables n should also be small (up to 10). However, this number can be also relatively large (10 to 100) provided that the problem data exhibits some kind of structured sparsity (to be specified later). Even though problem (1) is of independent interest, our initial motivation came from some applications in computer vision, where such problems are typical. These applications will be described elsewhere.

In such a situation, the fractional programming problem (1) is quite challenging. Indeed, we make *no* assumption on the polynomials p_i, q_i whereas even with a relatively small number of fractions and under convexity (resp. concavity) assumptions on p_i (resp. q_i), problem (1) is hard to solve, especially if one wants to compute the global minimum. In addition, we are interested in solving problem (1) *globally*, that is, we do not content ourselves with a *local* optimum satisfying first order optimality conditions, as is typically the case with standard local optimization algorithms such as Newton's method or its variants. If problem (1) is too difficult to solve globally (because of ill-conditioning and/or a large a number of variables or terms in the objective function), we would like to have at least a valid lower bound on the global minimum, since upper bounds can be obtained with local optimization algorithms.

In the literature, problem (1) is often called a sum-of-ratios program. Without assumptions, it is NP-complete [9]. Surveys on methods for solving this problem have been made in [23] and then [31]. The reader can especially refer to [31, Table 1]. According to this survey, many attempts make restrictive assumptions either on the concavity or linearity of the ratios. When the ratios are nonlinear, the most popular methods are based on Branch and Bound approaches, see e.g. [3]. Because this problem often occurs in computer vision, many practical studies have been carried out. These studies aim at solving problem (1) in the quadratic case [14], or at certifying a posteriori if a solution is global or not [10].

In this paper, we would like to approach the fractional programming problem (1) as a generalized moment problem solved with its hierarchy of convex semidefinite programming (SDP) relaxations:

- One possible approach is to reduce all fractions p_i/q_i to same denominator $\widetilde{q}$ and obtain a single rational fraction to minimize. Then one may try to apply the hierarchy of SDP relaxations defined in [13], see also [19, Section 5.8]. But such a strategy is not appropriate because the degree of $\widetilde{q}$ is potentially large and even if n is small, one cannot even implement the first relaxation of the hierarchy, due to present limitations of SDP solvers. Moreover, in general this strategy also destroys potential sparsity patterns present in the original formulation (1), and so precludes from using an appropriate version (for the rational fraction case) of the sparse semidefinite relaxations introduced in [28] whose convergence was proved in [17] under some conditions on the sparsity pattern, see also [19, Sections 4.6 and 5.3.4].

- Another possibility is to introduce additional variables r_i (that we may call lifting variables) with associated constraints

$$\frac{p_i(\mathbf{x})}{q_i(\mathbf{x})} \leq r_i,\ i = 1, \ldots, N,$$

and solve the equivalent problem:

$$f^* := \inf_{(\mathbf{x},\mathbf{r}) \in \widehat{\mathbf{K}}} \sum_{i=1}^{N} r_i \tag{3}$$

which is now a *polynomial* optimization problem in the new variables$(\mathbf{x}, \mathbf{r}) \in \mathbb{R}^n \times \mathbb{R}^N$, and where the new feasible set $\widehat{\mathbf{K}} = \mathbf{K} \cap \{(\mathbf{x}, \mathbf{r}) \in \mathbb{R}^{n+N} \ : \ r_i q_i(\mathbf{x}) - p_i(\mathbf{x}) \geq 0\}$ is modeling the epigraphs of the rational terms. This approach is an extension of the approach in [13] (it coincides if N = 1). Potential sparsity patterns in the $\mathbf{x}$ variables are preserved and if $\mathbf{K}$ is compact one may in general obtain upper and lower bounds $\overline{r}_i, \underline{r}_i$ on the r_i so as to make $\widehat{\mathbf{K}}$ compact by adding the quadratic (redundant) constraints $(r_i - \underline{r}_i)(\overline{r}_i - r_i) \geq 0$, $i = 1, \ldots, N$, and apply a sparse version of semidefinite relaxations [13]. However, in doing so one introduces N additional variables, and this may have an impact on the overall performance, especially if N is large. In the sequel this approach is referred to as the *epigraph approach*.

The goal of the present paper is to overcome the above difficulties in the following two situations: either n is relatively small, or n is relatively large but some sparsity is present, i.e., each f_i and each g_j in (1) is concerned with only a small subset of variables. In the approach that we propose, we do not need the epigraph liftings. The idea is to formulate (1) as an equivalent infinite-dimensional linear problem which a particular instance of the generalized moment problem (GMP) as defined in [18], with N unknown measures (where each measure is associated with a fraction p_i/q_i). In turn this problem can be easily modeled and solved with our public-domain software GloptiPoly 3 [12], a significant update of GloptiPoly 2 [11]. In the sequel this approach is referred to as the *GMP approach*. A collection of numerical experiments shows the potential of our GMP approach, especially in comparison with the epigraph approach. Numerical results also show that our results compare favorably with BARON software.

The outline of the paper is as follows. In Section 2 we introduce the SDP relaxations first in the case where n is small and with no sparsity pattern. Then in Section 3 we extend the SDP relaxations to the case that n is relatively large but some sparsity is present and can be exploited. In Section 4 we show how the GMP formulation can be exploited to model the SDP relaxations of problem (1) easily with GloptiPoly 3. The last section is devoted to a comparison with the epigraph approach and with the BARON solver on some numerical experiments.

2 Dense SDP relaxations

Consider the fractional programming problem (1) over the semi-algebraic set (2). We assume that $f^* > -\infty$ and the two polynomials:

$$\widetilde{p}(\mathbf{x}) := \sum_{i=1}^{N} \left(p_i(\mathbf{x}) \prod_{j \neq i}^{N} q_j(\mathbf{x}) \right) \tag{4}$$

$$\widetilde{q}(\mathbf{x}) := \prod_{i=1}^{N} q_i(\mathbf{x}), \tag{5}$$

are relatively prime. Hence, without loss of generality, $q_i > 0$ on $\mathbf{K}$ for each $i = 1, \ldots, N$. Indeed, by reducing the cost function to a common denominator, we obtain :

$$\sum_{i=1}^{N} f_i(\mathbf{x}) = \frac{\widetilde{p}(\mathbf{x})}{\widetilde{q}(\mathbf{x})}. \tag{6}$$

So, if $i_0 \in \mathbb{N}$ exists, such that q_{i_0} changes its sign on $\mathbf{K}$, then $\widetilde{q}$ changes its sign on $\mathbf{K}$. Thus, according to [13, Corollary 1 and Theorem 2], $f^* = -\infty$, which contradicts our hypothesis. Moreover, we assume that n, the number of variables in problem (1), is small, say up to 10.

2.1 GMP formulation

Consider the infinite dimensional linear problem

$$\begin{aligned} \hat{f} := \inf_{\mu_i \in \mathcal{M}(\mathbf{K})} & \sum_{i=1}^{N} \int_{\mathbf{K}} p_i \, d\mu_i \\ \text{s.t.} & \int_{\mathbf{K}} q_1 d\mu_1 = 1 \\ & \int_{\mathbf{K}} \mathbf{x}^\alpha q_i d\mu_i = \int_{\mathbf{K}} \mathbf{x}^\alpha q_1 d\mu_1, \quad \forall \alpha \in \mathbb{N}^n, \; i = 2, \ldots, N, \end{aligned} \tag{7}$$

where $\mathcal{M}(\mathbf{K})$ is the space of finite Borel measures supported on $\mathbf{K}$.

Theorem 2.1 *Let $\mathbf{K} \subset \mathbb{R}^n$ in (2) be compact, and assume that $q_i > 0$ on $\mathbf{K}$, $i = 1, \ldots, N$. Then $\hat{f} = f^*$.*

Proof: We first prove that $f^* \geq \hat{f}$. As $f = \sum_i p_i/q_i$ is continuous on $\mathbf{K}$, there exists a global minimizer $\mathbf{x}^* \in \mathbf{K}$ with $f(\mathbf{x}^*) = f^*$. Define $\mu_i := q_i(\mathbf{x}^*)^{-1} \delta_{\mathbf{x}^*}$, $i = 1, \ldots, N$, where $\delta_{\mathbf{x}^*}$ is the Dirac measure at $\mathbf{x}^*$. Then obviously, the measures (μ_i), $i = 1, \ldots, N$, are feasible for (7) with associated value

$$\sum_{i=1}^{N} \int_{\mathbf{K}} p_i d\mu_i = \sum_{i=1}^{N} p_i(\mathbf{x}^*)/q_i(\mathbf{x}^*) = f(\mathbf{x}^*) = f^*.$$

Conversely, let (μ_i) be a feasible solution of (7). For every $i = 1, \ldots, N$, let $d\nu_i$ be the measure $q_i d\mu_i$, i.e.

$$\nu_i(B) := \int_{\mathbf{K}\cap B} q_i(\mathbf{x}) d\mu_i(\mathbf{x})$$

for all sets B in the Borel σ-algebra of $\mathbb{R}^n$, and so the support of ν_i is $\mathbf{K}$. As measures on compact sets are moment determinate, see the Appendix, the moments constraints of (7) imply that $\nu_i = \nu_1$, for every $i = 2, \ldots, N$, and from $\int_{\mathbf{K}} q_1 d\mu_1 = 1$ we also deduce that ν_1 is a probability measure on $\mathbf{K}$. But then

$$\begin{aligned}\sum_{i=1}^N \int_{\mathbf{K}} p_i d\mu_i &= \sum_{i=1}^N \int_{\mathbf{K}} \frac{p_i}{q_i} q_i d\mu_i = \sum_{i=1}^N \int_{\mathbf{K}} \frac{p_i}{q_i} d\nu_1 \\ &= \int_{\mathbf{K}} \left(\sum_{i=1}^N \frac{p_i}{q_i}\right) d\nu_1 = \int_{\mathbf{K}} f d\nu_1 \geq \int_{\mathbf{K}} f^* d\nu_1 = f^*,\end{aligned}$$

where we have used that $f \geq f^*$ on $\mathbf{K}$ and ν_1 is a probability measure on $\mathbf{K}$. □

We next make the following assumption meaning that the set $\mathbf{K}$ admits analgebraic certificate of compactness.

Assumption 2.1 *The set $\mathbf{K} \subset \mathbb{R}^n$ in (2) is compact and the quadratic polynomial $\mathbf{x} \mapsto M - |\mathbf{x}|^2$ can be written as*

$$M - |\mathbf{x}|^2 = \sigma_0 + \sum_{j=1}^m \sigma_j \, g_j,$$

for some polynomials $\sigma_j \in \mathbb{R}[\mathbf{x}]$, all sums of squares of polynomials.

Remark 2.1 *If Assumption (2.1) is not satisfied (or cannot be checked easily) and if we know some $M > 0$ such that:*

$$\mathbf{K} \subset \left\{\mathbf{x} : \|\mathbf{x}\|^2 \leq M\right\}, \tag{8}$$

then it suffices to add the redundant quadratic constraint $M - \|\mathbf{x}\|^2 \geq 0$ in the definition of $\mathbf{K}$.

2.2 A hierarchy of dense SDP relaxations

Let $\mathbf{y}_i = (y_{i\alpha})$, $\alpha \in \mathbb{N}^n$, be a real sequence indexed in the canonical basis $(\mathbf{x}^\alpha)$ of $\mathbb{R}[\mathbf{x}]$, $i = 1, \ldots, N$, and for every $k \in \mathbb{N}$, let $\mathbb{N}^n_k := \{\alpha \in \mathbb{N}^n : \sum_j \alpha_j \leq k\}$.

Define the *moment* matrix $M_k(\mathbf{y}_i)$ of order k, associated with $\mathbf{y}$, whose entries indexed by multi-indices β (rows) and γ (columns) read

$$[M_k(\mathbf{y}_i)]_{\beta,\gamma} := y_{i(\beta+\gamma)}, \quad \forall\, \beta,\, \gamma \in \mathbb{N}^n_k,\, i = 1, \ldots, N,$$

and so are linear in $\mathbf{y}_i$. Similarly, given a polynomial $g(\mathbf{x}) = \sum_\alpha g_\alpha \mathbf{x}^\alpha$, define the *localizing* matrix $M_k(g\,\mathbf{y}_i)$ of order k, associated with $\mathbf{y}$ and g, whose entries read

$$[M_k(g\,\mathbf{y}_i)]_{\beta,\gamma} := \sum_\alpha g_\alpha y_{i(\alpha+\beta+\gamma)}, \quad \forall\, \beta, \gamma \in \mathbb{N}^n_k,\, i = 1, \ldots, N.$$

In particular, matrix $M_0(g\,\mathbf{y}_i)$ is identical to $L_{\mathbf{y}_i}(g)$ where for every i, $L_{\mathbf{y}_i}:\mathbb{R}[\mathbf{x}]\to\mathbb{R}$ is the Riesz linear functional defined by:

$$g \;\mapsto\; L_{\mathbf{y}_i}(g) \;:=\; \sum_{\alpha\in\mathbb{N}^n} g_\alpha y_{i\alpha}, \qquad \forall g\in\mathbb{R}[\mathbf{x}].$$

Let:

$$u_i := \lceil (\deg q_i)/2\rceil,\; i=1,\ldots,N,$$

$$r_j := \lceil (\deg g_j)/2\rceil,\; j=1,\ldots,m, \text{ and } r:=\max_j r_j,$$

and with no loss of generality assume that $u_1\le u_2\le\ldots\le u_N$. Consider the hierarchy of semidefinite programming (SDP) relaxations:

$$\begin{array}{rll} f_k^* = \inf\limits_{\mathbf{y}_i} & \displaystyle\sum_{i=1}^N L_{\mathbf{y}_i}(p_i) & \\ \text{s.t.} & M_k(\mathbf{y}_i)\succeq 0, & i=1,\ldots,N \\ & M_{k-r_j}(g_j\mathbf{y}_i)\succeq 0, & i=1,\ldots,N,\; j=1,\ldots,m \\ & L_{\mathbf{y}_1}(q_1)\;=\;1, & \\ & L_{\mathbf{y}_i}(\mathbf{x}^\alpha q_i) = L_{\mathbf{y}_1}(\mathbf{x}^\alpha q_1), & \forall\alpha\in\mathbb{N}^n_{2(k-u_i)},\; i=2,\ldots,N, \end{array} \tag{9}$$

with $2k\geqslant u_N$.

Theorem 2.2 *Let Assumption 2.1 hold and consider the hierarchy of SDP relaxations (9). Then it follows that*

(a) $f_k^*\uparrow f^*$ *as* $k\to\infty$.

(b) Moreover, let $(\mathbf{y}_i^k)$ *be an optimal solution of (9), such that*

$$\operatorname{rank} M_k(\mathbf{y}_i^k) \;=\; \operatorname{rank} M_{k-r}(\mathbf{y}_i^k) \;=:\; R, \quad i=1,\ldots,N.$$

if $R=1$ *then* $f_k^*=f^*$,

if $R>1$ *then each* $\mathbf{y}_i^k$ *has a representating measure* μ_i *supported on* R *points* $x^l(i)\in\mathbf{K}$, $l=1,\ldots,R$, *which can be extracted. If* $x^l(i)=x^l(j)$ *for all* $i\neq j$ *and every* l, *then* $f_k^*=f^*$.

Proof: Let us first prove statement (a). As $\mathbf{K}$ is compact, with no loss of generality we may and will assume that $\mathbf{K}\subset\{\mathbf{x}: M-\|\mathbf{x}\|^2\ge 0\}$ for some $M>0$, and we also assume that the quadratic redundant constraint $M-\|\mathbf{x}\|^2\ge 0$ constraint is part of the description of $\mathbf{K}$, modeled by the constraint $g_1(\mathbf{x})\ge 0$.

Then the semidefinite constraint $\mathbf{M}_{k-1}(g_1\,\mathbf{y}_i)\succeq 0$ implies

$$0\le\; L_{\mathbf{y}_i}(x_t^{2\ell}g_1(\mathbf{x})) \;=\; ML_{\mathbf{y}_i}(x_t^{2\ell}) - L_{\mathbf{y}_i}(x_t^{2\ell+2}) - \underbrace{\sum_{s\neq t}^n L_{\mathbf{y}_i}(\mathbf{x}_t^{2\ell}x_s^2)}_{\ge 0},$$

for all $0 \leq \ell \leq k-1$, and all $t = 1, \ldots, n$. Hence, an easy induction yields that for any feasible solution $\mathbf{y}_i$ one has:

$$M\, y_{i0} \geq L_{\mathbf{y}_i}(x_t^2);\; M^2\,(y_{i0}) \geq L_{\mathbf{y}_i}(x_t^4); \cdots ; M^k\, y_{i0} \geq L_{\mathbf{y}_i}(x_t^{2k}), \tag{10}$$

for every $t = 1, \ldots, n$.

Next by [19, Proposition 3.6], if $\mathbf{M}_k(\mathbf{y}_i) \succeq 0$ then

$$|y_{i\alpha}| \;\leq\; \max[y_{i0}, [\max_t\; L_{\mathbf{y}_i}(x_t^{2k})]\,], \qquad \forall \alpha \text{ with } |\alpha| \leq 2k. \tag{11}$$

Next as $q_1 > 0$ on $\mathbf{K}$ (compact) then $q_1 > \delta$ on $\mathbf{K}$ for some $\delta > 0$, and so by Putinar's Theorem [22] there exists k_0 such that

$$q_1 - \delta = \sigma_0 + \sum_{j=1}^{m} \sigma_j\, g_j,$$

for some SOS polynomials σ_j such that $\deg \sigma_j g_j \leq 2k$ for all j (with $g_0 = 1$). Applying $L_{\mathbf{y}_i}$ and using $L_{\mathbf{y}_i}(q_1) = 1$, yields:

$$1 - y_{i0}\,\delta \;=\; L_{\mathbf{y}_i}\left(\sum_{j=0}^{m} \sigma_j\, g_j\right) \;=\; \sum_{j=0}^{m} L_{\mathbf{y}_i}\,(\sigma_j\, g_j) \;\geq\; 0,$$

as soon as $k \geq k_0$. And so $y_{i0} \leq \delta^{-1}$, which in view of (10)-(11) yields

$$|y_{i\alpha}| \;\leq\; \max[\delta^{-1}, M^k \delta^{-1}] \;=:\; S_k, \quad \forall \alpha \text{ with } |\alpha| \leq 2k.$$

And in fact, a soon as $k \geq k_0$,

$$|y_{i\alpha}| \;\leq\; \max[\delta^{-1}, M^{|\alpha|} \delta^{-1}] \;=:\; S_{|\alpha|}, \quad \forall \alpha.$$

So for every $k \geq k_0$, the feasible set is compact which ensures the existence of an optimal solution $\mathbf{y}_i^k$.

So let $\mathbf{y}_i^k$ be a nearly optimal solution as in Theorem 2.2(a). Complete the finite sequence $\mathbf{y}_i^k$ with zeros to make it an infinite sequence indexed in $\mathbb{N}^n$ (instead of $\mathbb{N}^n_{2k}$). As soon as $k \geq k_0$ define the new sequence $\mathbf{w}_i^k = ((w_i^k)_\alpha)$, $\alpha \in \mathbb{N}^n$, by

$$w_{i\alpha}^k \;=\; y_{i\alpha}^k / S_{|\alpha|},$$

so that $|w_{i\alpha}^k| \leq 1$ for all α, and all $k \geq k_0$.

Hence every element $\mathbf{w}_i^k$ is an element of the unit ball $\mathbf{B}$ of the Banach space ℓ_∞ of infinite sequences, uniformly bounded. Equipped with the usual sup-norm, ℓ_∞ is the topological dual of ℓ_1. And so by Banach-Alaoglu's Theorem the unit ball of ℓ_∞ being compact (and sequentially compact) in the weak-$\star$ topology $\sigma(\ell_\infty, \ell_1)$, there exists a subsequence k_t and an infinite sequence $\mathbf{w}_i \in \mathbf{B}$ such that $\mathbf{w}_i^{k_t} \to \mathbf{w}_i$ as $t \to \infty$, for the weak-$\star$ topology, and in particular pointwise convergence holds, i.e.,

$$\lim_{t \to \infty} w_{i\alpha}^{k_t} \;=\; w_{i\alpha}, \qquad \forall \alpha \in \mathbb{N}^n.$$

Recalling the definition of $\mathbf{w}_i^k$, one may define an infinite sequence $\mathbf{y}_i$ by:

$$y_{i\alpha} \;:=\; w_{i\alpha}\, S_{|\alpha|}, \qquad \alpha \in \mathbb{N}^n,$$

to obtain:

$$\lim_{t\to\infty} y_{i\alpha}^{kt} \;=\; S_{|\alpha|}\lim_{t\to\infty} w_{i\alpha}^{kt} \;=\; S_{|\alpha|}w_{i\alpha} \;=:\; y_{i\alpha}, \qquad \forall \alpha\in\mathbb{N}^n.$$

Statement (b) follows from the flat extension theorem of Curto and Fialkow [7], see e.g. [19, Theorem 3.7] or [20, Section 5], so that each measure ν_i is supported on R points of $\mathbf{K}$. □

Remark 2.2 *Note that Theorem 2.2 is a generalization of [13, Theorem 9]. However, when N is large, [13, Theorem 9] is unsuitable to solve* (1)*. Indeed, the dual hierarchy of SDP relaxations used in [13, Theorem 9] is given by:*

$$\begin{array}{ll} \inf\limits_{\mathbf{y}} & L_{\mathbf{y}}(\widetilde{p}) \\ s.t. & L_{\mathbf{y}}(\widetilde{q}) \;=\; 1 \\ & M_k(\mathbf{y}) \succeq 0 \\ & M_{k-r_j}(g_j\mathbf{y}) \succeq 0, j=1,\ldots,m, \end{array} \tag{12}$$

with $2k \geqslant \deg(\widetilde{q})$ and $\widetilde{p}$, $\widetilde{q}$ defined by (6)*. If N is large, then the degree of $\widetilde{p}$ and $\widetilde{q}$ is large and the corresponding relaxations cannot be solved by conventional SDP solvers (see e.g. [6, 4, 32, 27, 25]).*

3 Sparse SDP relaxations

In this section we assume that the number n of variables in problem (1), is large, say from 10 to 100, and moreover we also assume that some sparsity pattern is present in the polynomial data.

3.1 GMP formulation

Let $I_0 := \{1,\ldots,n\} = \cup_{i=1}^N I_i$ with possible overlaps, and let $\mathbb{R}[x_k : k\in I_i]$ denote the ring of polynomials in the variables x_k, $k\in I_i$. Denote by n_i the cardinality of I_i.

One will assume that $\mathbf{K}\subset\mathbb{R}^n$ in (2) is compact, and one knows some $M>0$ such that $\mathbf{x}\in\mathbf{K} \Rightarrow M-|\mathbf{x}|^2\geq 0$. For every $i\leq N$, introduce the quadratic polynomial $\mathbf{x}\mapsto g_{m+i}(\mathbf{x}) = M - \sum_{k\in I_i} x_k^2$. The index set $\{1,\ldots,m+N\}$ has a partition $\cup_{i=1}^N J_i$ with $J_i\neq\emptyset$ for every $i=1,\ldots,N$. In the sequel we assume that for every $i=1,\ldots,N$, $p_i, q_i\in\mathbb{R}[x_k : k\in I_i]$ and for every $j\in J_i$, $g_j\in\mathbb{R}[x_k : k\in I_i]$. Next, for every $i=1,\ldots,N$, let

$$\mathbf{K}_i \;:=\; \{\mathbf{z}\in\mathbb{R}^{n_i} \;:\; g_k(\mathbf{z})\geq 0,\; k\in J_i\}$$

so that $\mathbf{K}$ in (2) has the equivalent characterization

$$\mathbf{K} \;=\; \{\mathbf{x}\in\mathbb{R}^n \;:\; (x_k, k\in I_i)\in\mathbf{K}_i,\; i=1,\ldots,N\}.$$

Similarly, for every $i,j\in\{1,\ldots,N\}$ such that $i\neq j$ and $I_i\cap I_j\neq\emptyset$,

$$\mathbf{K}_{ij} \;=\; \mathbf{K}_{ji} \;:=\; \{(x_k,\, k\in I_i\cap I_j) \;:\; (x_k,\, k\in I_i)\in\mathbf{K}_i;\; (x_k,\, k\in I_j)\in\mathbf{K}_j\,\}.$$

Let $\mathcal{M}(\mathbf{K})$ be the space of finite Borel measures on $\mathbf{K}$, and for every $i = 1, \ldots, N$, let $\pi_i : \mathcal{M}(\mathbf{K}) \to \mathcal{M}(\mathbf{K}_i)$ denote the projection on $\mathbf{K}_i$, that is, for every $\mu \in \mathcal{M}(\mathbf{K})$:

$$\pi_i\mu(B) \;:=\; \mu(\{\mathbf{x} \;:\; \mathbf{x} \in \mathbf{K};\, (x_k, k \in I_i) \in B\}), \quad \forall B \in \mathcal{B}(\mathbf{K}_i)$$

where $\mathcal{B}(\mathbf{K}_i)$ is the usual Borel σ-algebra associated with $\mathbf{K}_i$.

For every $i, j \in \{1, \ldots, N\}$ such that $i \neq j$ and $I_i \cap I_j \neq \emptyset$, the projection $\pi_{ij} : \mathcal{M}(\mathbf{K}_i) \to \mathcal{M}(\mathbf{K}_{ij})$ is also defined in an obvious similar manner. For every $i = 1, \ldots, N-1$ define the set:

$$U_i \;:=\; \{\, j \in \{i+1, \ldots, N\} \;:\; I_i \cap I_j \neq \emptyset \,\},$$

and consider the infinite dimensional problem

$$\begin{aligned} \hat{f} \;:=\; \inf_{\mu_i \in \mathcal{M}(\mathbf{K}_i)} \;& \sum_{i=1}^{N} \int_{\mathbf{K}_i} p_i \, d\mu_i \\ \text{s.t.} \;& \int_{\mathbf{K}_i} q_i d\mu_i = 1, \quad i = 1, \ldots, N \\ & \pi_{ij}(q_i d\mu_i) \;=\; \pi_{ji}(q_j d\mu_j), \quad \forall j \in U_i,\, i = 1, \ldots, N-1. \end{aligned} \tag{13}$$

Definition 3.1 *Sparsity pattern $(I_i)_{i=1}^N$ satisfies the running intersection property if for every $i = 2, \ldots, N$:*

$$I_i \bigcap \left(\bigcup_{k=1}^{i-1} I_k \right) \;\subseteq\; I_j, \qquad \textit{for some } j \leq i-1. \tag{14}$$

Remark 3.1 *: In general, the running intersection property (14) holds for sparsity patterns associated with many rational estimation problems in engineering (e.g. computer vision). Indeed, these problems are formulated as least squares problems where each residual is the difference between an observed parameter q_i and a rational function $f(x, r_i)$ recalibrated to reflect the observed behaviour:*

$$\min_{(x,r) \in \mathbb{R}^n \times \mathbb{R}^{lN}} \sum_{i=1}^{N} \|q_i - f(x, r_i)\|_2^2. \tag{15}$$

In this formulation, unknowns are $x \in \mathbb{R}^n$ and each $r_i \in \mathbb{R}^l$, $i = 1, \ldots, N$. The vector x represents the parameters of the model. The vector $r \in \mathbb{R}^{lN}$ represents the input observations. Classicaly, these variables are fixed. However, as is often the case in computer vision, it is necessary to reestimate r in order to increase the accuracy of the final model. Let

$$I_0 := \{ \underbrace{1, 2, \ldots, l}_{r_1}, \underbrace{l+1, \ldots, 2l}_{r_2}, \ldots, \underbrace{(N-1)l+1, \ldots, Nl}_{r_N}, \underbrace{Nl+1, \ldots, Nl+n}_{x} \}$$

be the set of indices which correspond to the variables $\{r_1, \ldots, r_N, x_1, \ldots, x_n\}$. Then, $I_0 := \bigcup_{i=1}^N I_i$ with $I_i = \{(i-1)l+1, \ldots, il, Nl+1, \ldots, Nl+n\}$ and, because

$$I_i \cap I_j = \{Nl+1, \ldots, Nl+n\} \quad \forall i \neq j,$$

the sparsity pattern $(I_i)_{i=1,\ldots,N}$ satisfies the running intersection property (14).

Theorem 3.1 *Let $\mathbf{K} \subset \mathbb{R}^n$ in (2) be compact. If the sparsity pattern $(I_i)_{i=1}^N$ satisfies the running intersection property (14) then $\hat{f} = f^*$.*

Proof: That $\hat{f} \leq f^*$ is straightforward. As $\mathbf{K}$ is compact and $q_i > 0$ on $\mathbf{K}$ for every $i = 1, \ldots, N$, $f^* = \sum_{i=1}^N f_i(\mathbf{x}^*)$ for some $\mathbf{x} \in \mathbf{K}$. So let μ be the Dirac measure $\delta_{\mathbf{x}^*}$ at $\mathbf{x}^*$ and let ν_i be the projection $\pi_i\mu$ of μ on $\mathbf{K}_i$. That is $\nu_i = \delta_{(x_k^*, k\in I_i)}$, the Dirac measure at the point $(x_k^*, k \in I_i)$ of $\mathbf{K}_i$. Next, for every $i = 1, \ldots, N$,define the measure $d\mu_i := q_i(\mathbf{x}^*)^{-1} d\nu_i$. Obviously, (μ_i) is a feasible solution of (13) because $\mu_i \in \mathcal{M}(\mathbf{K}_i)$ and $\int q_i d\mu_i = 1$, for every $i = 1, \ldots, N$, and one also has:

$$(x_k^*,\, k \in I_i \cap I_j) \;=\; \pi_{ij}\mu_i = \pi_{ji}\mu_j, \quad \forall j \neq i \,\text{such that}\, I_j \cap I_i \neq \emptyset.$$

Finally, its value satisfies

$$\sum_{i=1}^N \int_{\mathbf{K}_i} p_i d\mu_i \;=\; \sum_{i=1}^N p_i(\mathbf{x}^*)/q_i(\mathbf{x}^*) \;=\; f^*,$$

and so $\hat{f} \leq f^*$.

We next prove the converse inequality $\hat{f} \geq f^*$. Let (μ_i) be an arbitrary feasible solution of (13), and for every $i = 1, \ldots, N$, denote by ν_i the probability measure on $\mathbf{K}_i$ with density q_i with respect to μ_i, that is,

$$\nu_i(B) \;:=\; \int_{\mathbf{K}_i \cap B} q_i(\mathbf{x})\, d\mu_i(\mathbf{x}), \qquad \forall B \in \mathcal{B}(\mathbf{K}_i).$$

By definition of the linear program (13), $\pi_{ij}\nu_i = \pi_{ji}\nu_j$ for every couple $j \neq i$ such that $I_j \cap I_i \neq \emptyset$. Therefore, by [19, Lemma B.13] there exists a probability measure ν on $\mathbf{K}$ such that $\pi_i\nu = \nu_i$ for every $i = 1, \ldots, N$. But then

$$\begin{aligned}\sum_{i=1}^N \int_{\mathbf{K}_i} p_i\, d\mu_i = \sum_{i=1}^N \int_{\mathbf{K}_i} \frac{p_i}{q_i}\, d\nu_i \;&=\; \sum_{i=1}^N \int_{\mathbf{K}_i} \frac{p_i}{q_i}\, d\nu \\ &=\; \int_{\mathbf{K}} \left(\sum_{i=1}^N \frac{p_i}{q_i}\right) d\nu \geq f^*\end{aligned}$$

and so $\hat{f} \geq f^*$. □

3.2 A hierarchy of sparse SDP relaxations

Let $\mathbf{y} = (y_\alpha)$ be a real sequence indexed in the canonical basis $(\mathbf{x}^\alpha)$ of $\mathbb{R}[\mathbf{x}]$. Define the Riesz linear functional $L_\mathbf{y} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$, by:

$$f\left(= \sum_{\alpha \in \mathbb{N}^n} f_\alpha \mathbf{x}^\alpha\right) \;\mapsto\; \sum_{\alpha \in \mathbb{N}^n} f_\alpha y_\alpha, \qquad \forall f \in \mathbb{R}[\mathbf{x}].$$

For every $i = 1, \ldots, N$, let

$$\mathbb{N}^{(i)} \;:=\; \{\, \alpha \in \mathbb{N}^n \;:\; \alpha_k = 0 \text{ if } k \notin I_i \,\}; \quad \mathbb{N}^{(i)}_k \;:=\; \{\, \alpha \in \mathbb{N}^{(i)} \;:\; \sum_i \alpha_i \leq k \,\}.$$

An obvious similar definition of $\mathbb{N}^{(ij)}$ $(=\mathbb{N}^{(ji)})$ and $\mathbb{N}_k^{(ij)}$ $(=\mathbb{N}_k^{(ji)})$ applies when considering $I_j \cap I_i \neq \emptyset$.

For every $i = 1, \ldots, N$, let $\mathbf{y}_i = (y_{i\alpha})$ be a given sequence indexed in the canonical basis of $\mathbb{R}[\mathbf{x}]$. The sparse moment matrix $M_k(\mathbf{y}_i, I_i)$ associated with $\mathbf{y}_i$, has its rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$ of $\mathbb{R}[x_k : k \in I_i]$, and with entries:

$$M_k(\mathbf{y}_i, I_i)_{\alpha,\beta} = L_{\mathbf{y}_i}(\mathbf{x}^{\alpha+\beta}) = y_{i(\alpha+\beta)}, \qquad \forall\, \alpha, \beta \in \mathbb{N}_k^{(i)}.$$

Similarly, for a given polynomial $h \in \mathbb{R}[x_k : k \in I_i]$, the sparse localizing matrix $M_k(h\,\mathbf{y}_i, I_i)$ associated with $\mathbf{y}_i$ and h, has its rows and columns indexed in the canonical basis $(\mathbf{x}^\alpha)$ of $\mathbb{R}[x_k : k \in I_i]$, and with entries:

$$M_k(h\,\mathbf{y}_i, I_i)_{\alpha,\beta} = L_{\mathbf{y}_i}(h\,\mathbf{x}^{\alpha+\beta}) = \sum_{\gamma \in \mathbb{N}^{(i)}} h_\gamma\, y_{i(\alpha+\beta+\gamma)}, \qquad \forall\, \alpha, \beta \in \mathbb{N}_k^{(i)}.$$

With $\mathbf{K} \subset \mathbb{R}^n$ defined in (2), let $r_j := \lceil (\deg g_j)/2 \rceil$, for every $j = 1, \ldots, m+N$. Consider the hierarchy of semidefinite relaxations:

$$\begin{array}{rll} f_k^* = \inf\limits_{\mathbf{y}_1,\ldots,\mathbf{y}_N} & \sum\limits_{i=1}^N L_{\mathbf{y}_i}(p_i) & \\ \text{s.t.} & M_k(\mathbf{y}_i, I_i) \succeq 0, & i = 1, \ldots, N \\ & M_{k-r_j}(g_j \mathbf{y}_i, I_i) \succeq 0, & \forall j \in J_i,\ i = 1, \ldots, N \\ & L_{\mathbf{y}_i}(q_i) = 1, & i = 1, \ldots, N \\ & L_{\mathbf{y}_i}(\mathbf{x}^\alpha q_i) = L_{\mathbf{y}_j}(\mathbf{x}^\alpha q_j) = 0, & \forall \alpha \in \mathbb{N}^{(ij)},\ \forall j \in U_i,\ i = 1, \ldots, N-1 \\ & & \text{with } |\alpha| + \max[\deg q_i, \deg q_j] \leq 2k. \end{array} \tag{16}$$

Theorem 3.2 *Let $\mathbf{K} \subset \mathbb{R}^n$ in (2) be compact. Let the sparsity pattern $(I_i)_{i=1}^N$ satisfy the running intersection property, and consider the hierarchy of semidefinite relaxations defined in (16). Then:*

(a) $f_k^* \uparrow f^*$ *as* $k \to \infty$.

(b) *If an optimal solution $\mathbf{y}^* = (\mathbf{y}_1^*, \ldots, \mathbf{y}_N^*)$ of (16) satisfies*

$$\operatorname{rank} \mathbf{M}_k(\mathbf{y}_i^*, I_i) \;=\; \operatorname{rank} \mathbf{M}_{k-v_i}(\mathbf{y}_i^*, I_i) \;=:\; R_i, \qquad \forall i = 1, \ldots, N,$$

(where $v_i = \max_{j \in J_i}[r_j]$), and

$$\operatorname{rank} \mathbf{M}_k(\mathbf{y}_i^*, I_i \cap I_j) \;=\; 1, \qquad \forall j \in U_i,\ i = 1, \ldots, N-1,$$

then $f_k^* = f^*$ and one may extract finitely many global minimizers.

Proof: The proof is similar to that of Theorem 2.2 and also to that of [19, Theorem 4.7]. One first proves that (16) has a feasible solution and $f_k^* > -\infty$ for sufficiently large k. Next, let $\mathbf{y}^k = (\mathbf{y}_1^k, \ldots, \mathbf{y}_N^k)$ be a nearly optimal solution of (16), i.e.

$$f_k^* \leq \sum_{i=1}^N L_{\mathbf{y}_i^k}(p_i) \leq f_k^* + \frac{1}{k}.$$

(Again $\mathbf{K} \subset \{x : |x|^2 \leq M\}$ for some M because $\mathbf{K}$ is compact. So if such a scalar M is known, one may include the redundant quadratic constraint $M - |x|^2 \geq 0$ in the

definition of $\mathbf{K}$, in which case existence of an optimal solution to (16) is guaranteed.)then there exists a subsequence (k_ℓ) and a sequence $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$, such that

$$\lim_{\ell\to\infty} y_{i\alpha}^{k_\ell} \;=\; y_{i\alpha}, \quad \forall \alpha \in \mathbb{N}^{(i)}, i = 1, \ldots, N.$$

Of course, for every $i = 1, \ldots, N$, $\mathbf{y}_i$ is a sequence $(y_{i\alpha})$, $\alpha \in \mathbb{N}^n$, but only the entries $y_{i\alpha}$, $\alpha \in \mathbb{N}^{(i)}$, are relevant and all the other entries can be set to zero.

From this pointwise convergence it easily follows that for every $i = 1, \ldots, N$ and $j \in J_i$,

$$M_k(\mathbf{y}_i, I_i) \succeq 0,\; M_k(g_j\,\mathbf{y}_i, I_i) \succeq 0, \quad j \in J_i;\; i = 1, \ldots, N.$$

Now observe that each set $\mathbf{K}_i \subset \mathbb{R}^{n_i}$ satisfies Assumption 2.1. Therefore, by Putinar's theorem, see [22] or [19, Theorem 2.14], the sequence $\mathbf{y}^i = (y_{i\alpha})$, $\alpha \in \mathbb{N}^{(i)}$ (i.e., the subsequence of $\mathbf{y}_i$ ignoring the entries $y_{i\alpha}$, $\alpha \notin \mathbb{N}^{(i)}$), has a representing measure μ_i supported on $\mathbf{K}_i$. Again, by pointwise convergence, $L_{\mathbf{y}^i}(q_i)(= L_{\mathbf{y}_i}(q_i)) = 1$, for every $i = 1, \ldots, N$, and

$$L_{\mathbf{y}^i}(q_i\,\mathbf{x}^\alpha) \;=\; \int_{\mathbf{K}_i} \mathbf{x}^\alpha \underbrace{q_i(\mathbf{x})\,d\mu_i}_{d\nu_i(\mathbf{x})} \;=\; \int_{\mathbf{K}_j} \mathbf{x}^\alpha \underbrace{q_j(\mathbf{x})\,d\mu_j}_{d\nu_j(\mathbf{x})}, \quad \forall \alpha \in \mathbb{N}^{(ij)},\; \forall j \in U_i. \tag{17}$$

Therefore, for every $i = 1, \ldots, N$, $d\nu_i := q_i(\mathbf{x})d\mu_i$ is a finite Borel probability measure supported on $\mathbf{K}_i$. As measures on compact sets are moment determinate, see the Appendix, (17) yields:

$$\pi_{ij}\nu_i \;=\; \pi_{ji}\nu_j, \qquad \forall (i,j),\; j \in U_i.$$

Therefore, by [19, Lemma B.13] there exists a probability measure ν on $\mathbf{K}$ such that $\pi_i\nu = \nu_i$ for every $i = 1, \ldots, N$. But then

$$\begin{aligned} f^* \geq \lim_{\ell\to\infty} f_{k_\ell}^* &= \lim_{\ell\to\infty} \sum_{i=1}^N L_{\mathbf{y}_i^{k_\ell}}(p_i) = \sum_{i=1}^N \int_{\mathbf{K}_i} p_i\,d\mu_i \\ &= \sum_{i=1}^N \int_{\mathbf{K}_i} \frac{p_i}{q_i}\,q_i d\mu_i = \sum_{i=1}^N \int_{\mathbf{K}_i} \frac{p_i}{q_i}\,d\nu_i \\ &= \int_{\mathbf{K}} \left(\sum_{i=1}^N \frac{p_i}{q_i}\right) d\nu \geq f^*. \end{aligned}$$

As the converging subsequence was arbitrary, and (f_k^*) is monotone non decreasing, we finally get $f_k^* \uparrow f^*$. In addition, ν is an optimal solution of (13) with optimal value $f^* = \hat{f}$.

The proof of (b) is as in [17] and uses the flat extension theorem of Curto and Fialkow, see [7] or [19, Theorem 3.7], from which, the sequence $(\mathbf{y}^*)^i = (y_{i\alpha}^*)$, $\alpha \in \mathbb{N}^{(i)}$, has an atomic representing measure supported on R_i points of $\mathbf{K}_i$, for every $i = 1, \ldots, N$. □

4 GloptiPoly and examples

In this section we show that the generalized moment problem (GMP) formulation of rational optimization problem (1) has a straightforward Matlab implementation when using our software GloptiPoly 3 [12]. Rather than explaining the approach in full generality with awkward notations, we describe three simple examples.

4.1 Illustrative univariate examples

4.1.1 Wilkinson-like rational function

Consider the elementary univariate rational optimization problem

$$f^* = \sup_{x\in\mathbb{R}} f(x), \quad f(x) = \sum_{i=1}^{N} \frac{p_i(x)}{q_i(x)} = \sum_{i=1}^{N} \frac{1}{x^2+i}$$

with N an integer. The only real critical point is $x = 0$, at which the objective function takes its maximum

$$f^* = f(0) = \sum_{i=1}^{N} \frac{1}{i}.$$

Reducing to the same denominator

$$f(x) = \frac{\sum_i \prod_{j\neq i}(x^2+j)}{\prod_i (x^2+i)} = \frac{p(x^2)}{q(x^2)}$$

yields the well-known Wilkinson polynomial q whose squared root moduli are the integers from 1 to N. This polynomial was described in the mid 1960s by J.H. Wilkinson to illustrate the difficulty of computing (numerically) roots of polynomials. If we choose e.g. $N = 20$, reduction to the same denominator is hopeless since the constant coefficient in monic polynomial q is $20! = 2432902008176640000$. The GMP formulation (7) of this problem reads (up to replacing "inf" with "sup" in the objective function):

$$\begin{aligned} \sup_{\mu_i\in\mathcal{M}(\mathbb{R})} \quad & \sum_{i=1}^{N} \int_{\mathbb{R}} p_i d\mu_i \\ \text{s.t.} \quad & \int_{\mathbb{R}} q_1 d\mu_1 = 1 \\ & \int_{\mathbb{R}} x^\alpha q_i d\mu_i = \int_{\mathbb{R}} x^\alpha q_1 d\mu_1, \quad \forall \alpha \in \mathbb{N}^n,\ i = 1,\ldots,N. \end{aligned}$$

Our Matlab script to model and solve this problem is given in Appendix 7.1.

At the first SDP relaxation we obtain a rank-one moment matrix corresponding to a Dirac at $x^* = 0$ with $f^* = 3.5977$ which is consistent with Maple's output:

```
> f := sum(1/(x^2+i), i=1..20);
> evalf(subs(x = 0, f));
    3.5977
```

Note that for this example Assumption 2.1 is violated, since we optimize over the unbounded set $\mathbf{K} = \mathbb{R}$. In spite of this, we could solve the problem globally because for *univariate* problems, compactness of the feasible set is not required.

4.1.2 Relevance of the compactness assumption

With the next elementary example we would like to emphasize the practical impact of optimization on a bounded set $\mathbf{K}$ (and the smaller, the better). Consider the univariate

problem

$$f^* = \inf_{x \in \mathbf{K}} f(x), \quad f(x) = \frac{1+x+x^2}{1+x^2} + \frac{1+x^2}{1+2x^2}. \tag{18}$$

First let $\mathbf{K} = \mathbb{R}$. The numerator of the gradient of $f(x)$ has two real roots, one of which being the global minimum located at $x^* = -1.4215$ for which $f^* = 1.1286$. The GloptiPoly script given in Appendix 7.2 models and solves the SDP relaxations of orders $k = 0, \ldots, 9$ of the GMP formulation of (18). We retrieve the following monotonically increasing sequence of lower bounds f_k^* (up to 5 digits) obtained by solving the SDP relaxations (9):

order k	bound f_k^*	order k	bound f_k^*
1	1.0000	4	1.0958
2	1.0001	5	1.1285
3	1.0169	6	1.1286

Table 1: Lower bounds for SDP relaxations of problem (18).

At SDP relaxation $k = 9$, GloptiPoly certifies global optimality and extracts the global minimizer. Table 1 shows that the convergence of the hierarchy of SDP relaxations is rather slow for this very simple example. This is due to the fact that one optimizes over the unbounded set $\mathbf{K} = \mathbb{R}$. In Figure 1 we display the sequences of lower bounds obtained by solving the SDP relaxations of problem (18) on compact sets $\mathbf{K} = [-R,\ R]$ for $R = 2, 3, \ldots, 10$. Clearly, the smaller is R the faster is the convergence.

4.2 Exploiting sparsity with GloptiPoly

Even though version 3 of GloptiPoly may exploit problem sparsity, there is no illustration of this feature in the software user's guide [12]. In this section we provide such a simple example. Note also that GloptiPoly does *not* detect sparsity for a given problem, in contrast with SparsePOP which includes a routine to detect structural sparsity patterns by using a heuristic to build up chordal extensions of some graph [29] associated with the problem data. However, SparsePOP is not designed to handle directly rational optimization problems.

Consider the elementary example of [17, Section 3.2]:

$$\begin{array}{ll} \inf\limits_{\mathbf{x} \in \mathbb{R}^4} & x_1x_2 + x_1x_3 + x_1x_4 \\ \text{s.t.} & x_1^2 + x_2^2 \leq 1 \\ & x_1^2 + x_3^2 \leq 2 \\ & x_1^2 + x_4^2 \leq 3 \end{array}$$

for which the variable index subsets $I_1 = \{1, 2\}$, $I_2 = \{1, 3\}$, $I_3 = \{1, 4\}$ satisfy the running intersection property (14) in Definition 14. Note that this problem is a particular case of (1) with a polynomial objective function.

Without exploiting sparsity, the GloptiPoly script to solve this problem is given in Appendix 7.3.1. GloptiPoly certifies global optimality with a moment matrix of size 15, and

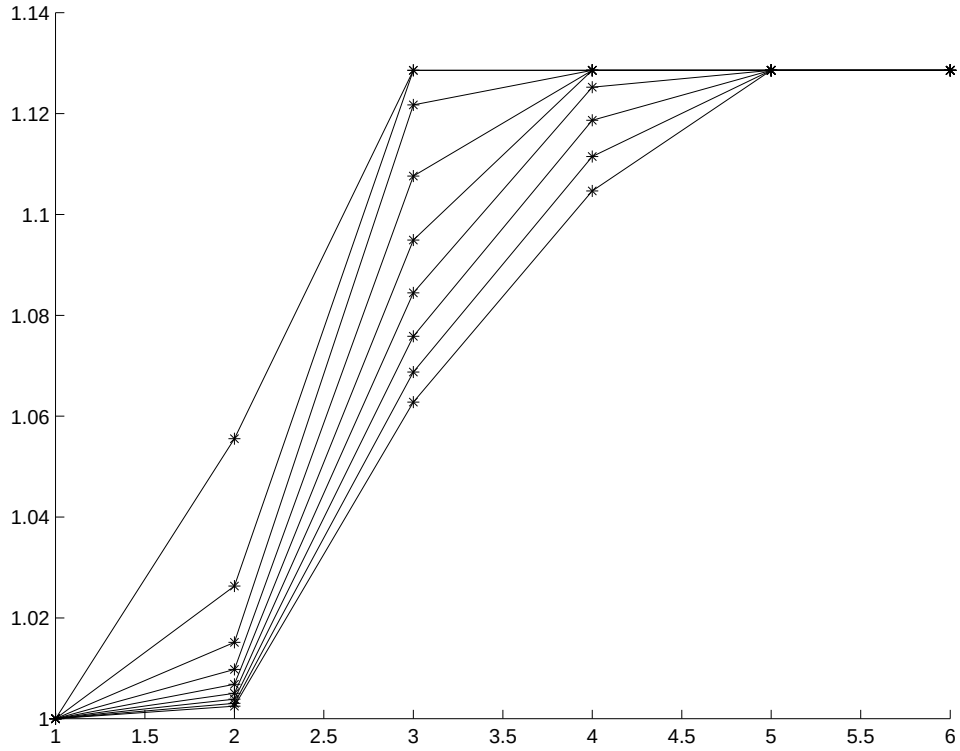

Figure 1: Lower bounds for SDP relaxations of problem (18) on bounded sets $K = [-R, R]$ for R=2 (top curve), 3, 4, 5, 6, 7, 8, 9 and 10 (bottom curve).

3 localizing matrices of size 5. The script exploiting sparsity, given in Appendix 7.3.2, associates variables with several measures μ_i, in a manner consistent with subsets I_i. GloptiPoly certifies global optimality with 3 moment matrices of size 6, and 3 localizing matrices of size 3.

Remark 4.1 *GloptiPoly is a Matlab interface which constructs LMI relaxations of a given GMP. The corresponding LMI relaxation can then be solved via any SDP solver interfaced via YALMIP [33]. So, for the same optimization problem converted by Gloptipoly, the number of function evaluations and the number of iterations depend on the solver used. For this reason, we believe that the latter two quantities cannot be considered as representative values in this context. However, the number of function evaluations and the number of iterations are directly related to the size of the problem converted by GloptiPoly. This size is represented by the sizes of moment matrices and localizing matrices.*

4.3 Comparison with the epigraph approach

Recall the epigraph optimization problem:

$$f^* := \inf_{(\mathbf{x},\mathbf{r})\in\widehat{\mathbf{K}}} \sum_{i=1}^{N} r_i \tag{3}$$

with:

$$\widehat{\mathbf{K}} \;=\; \mathbf{K} \cap \{(\mathbf{x},\mathbf{r}) \in \mathbb{R}^{n+N} \;:\; r_i q_i(\mathbf{x}) - p_i(\mathbf{x}) \geq 0\}.$$

Note that the epigraph approach is consistent with the running intersection property (14). Indeed, let $I_0 = \{1, 2, \ldots, n+N\}$ be the set of indices which correspond to the variables $\{r_1, \ldots, r_N, x_1, \ldots, x_n\}$. Then, $I_0 := \bigcup_{i=1}^{N} I_i$ with $I_i = \{i, N+1, \ldots, N+n\}$ and, because $I_i \cap I_j = \{N+1, \ldots, N+n\}\ \forall i \neq j$, the sparsity pattern $(I_i)_{i=1,\ldots,N}$ verifies the running intersection property. Consequently (see e.g. [19, Theorem 5.9]), if k is the relaxation order, the size of the moment matrix can be reduced from $\binom{n+N+k}{n+N}$ to $\binom{n+1+k}{n+1}$. Thus, in the sequel, we use this sparsity pattern in the epigraph approach to reduce the number of variables of the SDP relaxation.

In most of the examples that we have processed, the epigraph approach described in the Introduction (consisting of introducing one lifting variable for each rational term in the objective function) was less efficient than the GMP approach. Typically, the order of the SDP relaxation (and hence its size) required to certify global optimality is larger with the epigraph approach than with the GMP approach.

When testing the epigraph approach, we have also observed that numerically, it is better to replace the inequality constraints $r_i q_i(\mathbf{x}) - p_i(\mathbf{x}) \geq 0$ with equality constraints $r_i q_i(\mathbf{x}) - p_i(\mathbf{x}) = 0$ in the definition of semi-algebraic set $\mathbf{K}$ in (3). For the example of Section 4.1.1 the epigraph approach with inequalities certifies global optimality at order $k = 5$, whereas the epigraph approach with equalities requires $k = 1$.

Note that for all problems presented in this subsection, there is no sparsity pattern to exploit because all variables appear in each term of the sum.

4.3.1 Eighth-degree function

As a typical illustration of the issues faced with the epigraph approach,consider the example with eighth-degree terms

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^2} & f(x) := \sum_{i=1}^{10} \dfrac{(x_1+x_2)(x_1^2+x_1^2x_2^2+x_2^4+i^2) - (ix_2^2+1)(x_1^4+x_2^2+2i)}{(x_1^4+x_2^2+2i)(x_1^2+x_1^2x_2^2+x_2^4+i^2)}, \\ \text{s.t.} & x_i^2 \leqslant 10 \end{array} \tag{19}$$

which is cooked up to have several local optima and sufficiently high degree to prevent reduction to the same denominator. The GMP approach yields a certificate of global optimality with $x_1^* = -0.60490$, $x_2^* = -2.2053$, $f^* = -6.2844$ at order $k = 6$ in a few seconds on a standard PC. In contrast, the epigraph approach does not provide a certificate for an order as high as $k = 7$, and requires more than one minute of CPU time.

4.3.2 Kowalik's function

In order to illustrate this behaviour, the two approaches are compared on the Kowalik problem [1]. This academic problem is a minimization in the least squares sense, given by:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^2} & f(x) := \sum\limits_{k=1}^{11} \left(a_k - \dfrac{x_1(1+x_2b_k)}{(1+x_3b_k+x_4b_k^2)} \right)^2, \\ \text{s.t.} & (x_i - 0.21)^2 \leqslant 0.21^2 \end{array} \tag{20}$$

with :

$$\begin{aligned} a &= (0.195\ \ 0.194\ \ 0.173\ \ 0.160\ \ 0.084\ \ 0.062\ \ 0.045\ \ 0.034\ \ 0.032\ \ 0.023\ \ 0.024)\,, \\ b &= \big(\ 0.25\ \ \ 0.5\ \ \ 1\ \ \ 2\ \ \ 4\ \ \ 6\ \ \ 8\ \ \ 10\ \ \ 12\ \ \ 14\ \ \ 16\ \big)\,. \end{aligned}$$

The GMP approach yields a certificate of global optimality at order $k = 3$ in 10.7 seconds on a standard PC. The obtained solution matches with the known global minimizer to four significant digits. In contrast, the epigraph approach does not provide a certificate within a reasonable time. To obtain an approximation of the global minimizer, the search domain is reduce to the ball $\{x \in \mathbb{R}^4 : \sum_{i=1}^n (x_i - 0.21)^2 \leqslant 0.21\}$ which contains the global solution certified by the GMP approach. Then, as the global minimizer is unique, the vector of first order moments provides an approximation of the global minimizer. In fact this vector converges to the unique optimal solution (see [24]). Table 2 shows the certified solution obtained by the GMP approach and the approximation given by the minimization of the epigraph problem on the ball $\{x \in \mathbb{R}^4 : \sum_{i=1}^n (x_i - 0.21)^2 \leqslant 0.21\}$. In addition to high computational times, Table 2 also shows that the resulting approximation is far from the global minimizer. Note that to ensure the convergence of the sequence of

Method	x_1^*	x_2^*	x_3^*	x_4^*	f^*	times (s)	order k
GMP	0.192	0.190	0.123	0.135	3.10×10^{-4}	10	3
Epigraph	0.194	0.188	0.129	0.132	3.12×10^{-4}	208	3

Table 2: Solution certified by the GMP approach at $k = 3$ versus the approximation given by the minimization of the epigraph problem on the ball $\{x \in \mathbb{R}^4 : \sum_{i=1}^n (x_i - 0.21)^2 \leqslant 0.21\}$ at $k = 3$.

the solutions of the SDP relaxations of the epigraph formulation to the global minimizer, the lifting variables must be bounded. For solving problem (20), the lifting variables were bounded by using the following constraints:

$$r_k^2 \leqslant \left(a_k - \frac{x_1^*(1 + x_2^* b_k)}{(1 + x_3^* b_k + x_4^* b_k^2)} \right)^2,\ k = 1, \ldots, 11. \tag{21}$$

Consequently, an additional information is given to the epigraph approach (but not given in the GMP approach). However, even with this additional information, the GMP approach still provides better results.

4.3.3 De Jong's functions

Consider the modified De Jong's rational function minimization problem [21], given by:

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & f(x) := -\sum_{i=1}^{N} \frac{1}{(10x_1 + a_i)^2 + (10x_2 + b_i)^2 + \dfrac{c}{i^2}} \\ \text{s.t.} \quad & x_1^2 + (x_2 + 0.1)^2 \leqslant 0.85 \end{aligned} \tag{22}$$

with:

$$\begin{aligned} a &= (-4 \quad 0 \quad 4 \quad -4 \quad 0 \quad 4 \quad -4 \quad 0 \quad 4 \quad -4 \quad 0 \quad 4 \quad -4 \quad 0 \quad 4), \\ b &= (-4 \quad -4 \quad -4 \quad -1 \quad -1 \quad -1 \quad 2.5 \quad 2.5 \quad 2.5 \quad 6 \quad 6 \quad 6 \quad 9 \quad 9 \quad 9), \\ c &= 14. \end{aligned} \tag{23}$$

N	x_1^*	x_2^*	f^*	order k	times (s)
9	-0.4000	-0.2500	-6.037	6	4.40s
12	-0.4000	-0.6000	-10.56	8	11.4s
15	-0.4000	-0.9000	-16.40	13	81.7s

Table 3: Solutions, certification orders and computational times given by the GMP approach for problem (22).

The certified solutions obtained by the GMP approach are given in Table 3. As we already mentioned, the lifting variables of the epigraph formulation must be bounded. In most examples that we have processed, these bounds play an important role in the accuracy of the algorithm. If the bounds are properly set, the convergence speed and the accuracy of the solutions can be equivalent to that of the GMP approach. However, finding tight bounds on each lifting variable is a difficult problem a priori. In practice, a simple way to proceed is to bound each lifting variable thanks to the following constraints:

$$r_i^2 \leqslant \left(\frac{\max\limits_{x\in\mathbf{K}} |p_i(x)|}{\min\limits_{x\in\mathbf{K}} q_i(x)} \right)^2 , \ \forall i = 1 \dots N. \tag{24}$$

For problem (22), these constraints reduce to:

$$r_i^2 \leqslant \frac{i^4}{c^2}, \ \forall i = 1 \dots N. \tag{25}$$

For $N = 9, 12, 15$ we observe the same behaviour: the epigraph approach never provides a certified solution. However, because the global minimizer is unique, again the vector of first order moments provides an approximation of the global minimizer. Table 4 summarizes the approximations given by the epigraph approach, the orders k of the SDP relaxation from which the vector of first order moments is read, and the computational times required to solve this relaxation. This table shows off a significant gap between the approximation given by the epigraph approach and the certified solution given by the GMP approach. Note that, due to the increasing size of the relaxations, we have observed numerical instabilities of the SDP solver SeDuMi [25] used in Gloptipoly. These instabilities were observed for $k > 3$ with $N = 9$ and $k > 2$ with $N = 12, 15$.

N	x_1^*	x_2^*	f^*	order k	times (s)
9	-0.3484	-0.2417	-2.527	3	5.8s
12	-0.0272	-0.2887	-2.739	2	3.2s
15	-0.0323	-0.5588	-3.115	2	4.9s

Table 4: Approximation, orders and computational times given by the epigraph approach for problem (22).

4.4 Comparison with BARON solver

In the previous examples, we have compared the GMP approach with the epigraph approach. However, it is also possible to solve directly this particular class of problems by

using standard software packages. Among these methods, a reviewer of a previous version of this paper suggested to compare the GMP approach with BARON software [26] publicly available on the NEOS server [8]. We used the version 14.0.3 (built: LNX-64) of BARON available on NEOS, and the GMP experiments were run on our own computer with GloptiPoly version 3.7 on Matlab 2014 on an IntelCore i7 2.2GHz (4 cores) with 16GB RAM.

4.4.1 Shekel's foxholes

In order to compare the GMP approach with the BARON solver on some benchmark problems, consider the modified Shekel foxholes minimization problem [5]:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & f(x) := -\sum\limits_{i=1}^{N} \dfrac{1}{\sum_{j=1}^{n}(x_j - a_{ij})^2 + c_i} \\ \text{s.t.} & \sum\limits_{i=1}^{n}(x_i - 5)^2 \leqslant 60 \end{array} \tag{26}$$

whose data a_{ij}, c_i, $i = 1,\ldots,N$, $j = 1,\ldots,n$ are given. These functions have been designed to have many local minima. Note that it is not possible to exploit problem sparsity in this case, since all variables appear in each term of the sum (26). We know the unique global minima for the set of data [1, 16]. For this set, N is fixed to 30. Note that, in order to avoid numerical problems, a suitable scaling is done to make critical points fit within the ball $\{x \in \mathbb{R}^n : \sum_{i=1}^{n}(x_i - 0.5)^2 \leqslant 0.6\}$.

In the case $n = 5$, the GMP approach yields a certificate of global optimality at order $k = 3$ in about 84 seconds on a standard PC. The extracted minimizer is:

$$x^*_{GMP} = (8.025, 9.151, 5.114, 7.621, 4.564), \tag{27}$$

which matches with the known unique global minimizer to four significant digits. This point can be refined if given an initial guess for a local optimization method: if we use a standard quasi-Newton BFGS algorithm, we obtain after a few iterations a point matching the known unique global minimizer to eight significant digits. Comparatively, BARON provides an approximation of the unique global minimum with four correct significant digits in 1000 seconds which by default is set as a time limit on the NEOS server. The solver interrupts because the limit time is reached, with a solver status equal to 3. Consequently, the obtained approximation is not certified and detected as a local minimum. Moreover, if this limit is decreased to 100 seconds, the approximation remains equal to the unique global minimum with four significant digits but always detected as a local minimum. To obtain a certified solution, the time limit must be increased to approximately 2500 seconds.

In the case $n = 10$, for which the global minimum is given in [1, Table 17], the GMP approach yields a certificate of global optimality at order $k = 2$ in about 284 seconds of CPU time. Here too, we observe that the extracted minimizer

$$x^*_{GMP} = (8.025, 9.151, 5.114, 7.621, 4.564, 4.711, 2.996, 6.126, 0.7341, 4.982)$$

is a good approximation to the minimizer, with four correct significant digits. If necessary, this point can be used as an initial guess in some (local) optimization algorithm.

Comparatively, BARON provides an approximation of the unique global minimum with four correct significant digits in 3600 seconds which we set as a time limit on the NEOS server. The solver interrupts because the limit time is reached, with a solver status equal to 3. Consequently, the obtained approximation is not certified and detected as a local minimum. Moreover, if this limit is decreased to 300 seconds, the approximation remains equal to the unique global minimum with four significant digits but always detected as a local minimum.

4.4.2 Rosenbrock's functions

Consider the rational optimization problem

$$f^* = \max_{x\in\mathbb{R}^n} \sum_{i=1}^{n-1} \frac{1}{100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 + 1} \tag{28}$$

which has the same critical points as the well-known Rosenbrock problem

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

whose geometry is troublesome for local optimization solvers. Note that a sparsity pattern can be exploited to reduce the number of variables in the SDP relaxations. It can been easily shown that the global maximum $f^* = n-1$ of problem (28) is achieved at $x_i^* = 1$, $i = 1, \ldots, n$. Our experiments with local optimization algorithms reveal that standard quasi-Newton solvers or functions of the Optimization toolbox for Matlab, called repeatedly with random initial guesses, typically yield local maxima quite far from the global maximum.

With our GMP approach, after exploiting sparsity and adding bound constraints $x_i^2 \leq 16$, $i = 1, \ldots, n$, we could solve problem (28) with a certificate of global optimality for n up to 1000. A certificate of global optimality is obtained at order $k = 1$ with typical CPU times ranging from 10 seconds for $n = 100$ to 500 seconds for $n = 1000$.

By varying n from 20 to 1000 and submitting the corresponding problems to BARON on the NEOS solver, we observe the following behavior. First, BARON finds a solution (i.e. with a termination status equal to 1) with typical CPU times ranging from less than one second to 500 seconds. Second, BARON finds the global maximum in most of cases except for the values of n summarized on Table 5. Moreover, for $n \geqslant 640$ (except for $n = 860$), all solutions provided by BARON are wrong since all satisfy $x_1 = -0.995$ ($\neq 1$) and $x_k = 1$ for all $k \neq 1$ (the expected values of the global maximum). Consequently, it may be seen that problem (28) creates numerical instabilities which can be overcome thanks to GMP sparse formulation.

In order to increase the difficulty of problem (28), we defined the following rational optimization problem:

$$f^* = \max_{x_i^2 \leq 1} \sum_{i=1}^{n-1} \frac{(-1)^i}{100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 + 1}. \tag{29}$$

In the case $n = 17$, the GMP approach yields a certificate of global optimality at order $k = 8$ in about 26 seconds on a standard PC. The extracted minimizer is :

$$\begin{aligned} x^*_{GMP} = \; & (-0.006 \quad 0.839 \quad 0.703 \quad 1.00 \quad 1.00 \quad 0.667 \quad 0.444 \quad 0.844 \quad 0.711 \\ & 1.00 \quad 1.00 \quad 0.667 \quad 0.444 \quad 0.844 \quad 0.711 \quad 1.00 \quad 1.00 \quad -1.00), \end{aligned} \tag{30}$$

n	f^*	f^*_{BAR}
260	259	243.474
280	279	252.753
340	339	336.403
420	419	414.606

Table 5: Values of n for which BARON finds a solution far from the global maximum of problem (22), and the corresponding optimal values f^*_{BAR}.

and $f^*_{GMP} = 7.383$. Comparatively, in less than one second BARON finds (with a solver status equal to 1) the following solution:

$$x^*_{BAR} = \begin{array}{l} (-0.006 \quad 0.783 \quad 0.612 \quad 0.803 \quad 0.643 \quad 0.808 \quad 0.652 \quad 0.811 \quad 0.656 \\ \quad 0.815 \quad 0.663 \quad 0.827 \quad 0.682 \quad 0.864 \quad 0.746 \quad 1.000 \quad 1.000 \quad -1.000). \end{array} \tag{31}$$

which is not the correct global minimizer as $f^*_{BAR} = 7.342 < f_{GMP}$.

4.4.3 Kowalik's function

We revisit the Kowalik problem of Example 4.3.2. As reported in Table 4.4.3, BARON finds a good approximate solution in less than 2 seconds, whereas the GMP approaches certifies globally optimality of a solution which is slightly closer to the theoretical optimum in approx. 10 seconds.

Method	x^*_1	x^*_2	x^*_3	x^*_4	f^*	times (s)	order k
GMP	0.192	0.190	0.123	0.135	3.10×10^{-4}	10	3
BARON	0.192	0.184	0.113	0.134	3.12×10^{-4}	2	
Theoretical	0.192	0.190	0.123	0.135	3.10×10^{-4}		

4.4.4 Sum of linear and quadratic ratios

We have compared the performance of GMP and BARON on several instances of problem (1) with a small number of variables, found in the technical literature:

- [31, Example 5.1] is an instance with a sum of $N = 16$ ratios with linear numerators and denominators in $n = 3$ variables, on a polytopic set K. Our GMP approach certifies the global solution in approx. 14 seconds, while BARON solves this problem is less than 1 second.
- [31, Example 6.3], also reported in [3], is an instance with a sum of $N = 2$ ratios with quadratic numerators and denominators in $n = 3$ variables, on a polytopic set K. Both our GMP approach and BARON solve this problem is less than 1 second.
- [2, Example 6.2] is an instance with a sum of $N = 2$ ratios with quadratic numerators and denominators in $n = 2$ variables, on a polytopic set K. Both our GMP approach and BARON solve this problem is less than 1 second.

- [16, Example 5.1] is an instance with a sum of $N = 3$ ratios with linear numerators and denominators in $n = 3$ variables, on a polytopic set K. Our GMP approach solves the problem is approx. 2 seconds, whereas BARON solves this problem is less than 1 second.

- [16, Example 5.2] is an instance with a sum of $N = 4$ ratios with linear numerators and denominators in $n = 3$ variables, on a polytopic set K. Our GMP approach solves the problem is approx. 3 seconds, whereas BARON solves this problem is less than 1 second.

- [30, Example 5.1] is an instance with a sum of $N = 2$ ratios with quadratic numerators and denominators in $n = 3$ variables, on a quartic set K. Both our GMP approach and BARON solve this problem is less than 1 second.

- [30, Example 5.3] is an instance with a sum of $N = 2$ ratios with quadratic numerators and denominators in $n = 2$ variables, on a polytopic set K. Both our GMP approach and BARON solve this problem is less than 1 second.

5 Conclusion

The problem of minimizing a sum of *many* low-degree (typically non-convex) rational fractions on a (typically non-convex) semi-algebraic set arises in several important applications, and notably in computer vision (triangulation, estimation of the fundamental matrix in epipolar geometry) and in systems control (H_2 optimal control with a fixed-order controller of a linear system subject to parametric uncertainty). These engineering problems motivated our work, but the application of our techniques to computer vision and systems control will be described elsewhere. These fractional programming problems being non convex, local optimization approaches yield only upper bounds on the optimum.

In this paper we were interested in computing the global minimum (and possibly global minimizers) or at least, computing valid lower bounds on the global minimum, for fractional programs involving a sum with many terms. We have used a moment-SOS approach with semidefinite relaxations by formulating the rational optimization problem as an instance of the generalized moment problem (GMP). In addition, when the number of variables is large, some structured sparsity pattern in problem data can sometimes be exploited. In numerical experiments, our public-domain software GloptiPoly interfaced with off-the-shelf semidefinite programming solvers indicate that the approach can solve problems that can be challenging for state-of-the-art global optimization algorithms. This is consistent with the experiments made in [15] where the (dense) SDP relaxation approach was first applied to (polynomial) optimization problems of computer vision.

For larger and/or ill-conditioned problems, it can happen that GloptiPolyextracts from the moment matrix a minimizer which is not very accurate. It can also happen that GloptiPoly is not able to extract a minimizer, but when the global minimizer is unique (which is generically true) the vector of first-order moments (of an optimal solution of the relaxation) provides an approximation that converges to the global minimizer. And so this resulting approximation may be used as initial guess for any local optimization algorithm.

A comparison of our approach with other global optimization techniques (e.g., reported in Hans Mittelmann's or Arnold Neumaier's webpages) is out of the scope of this paper. We believe however that such a comparison would be fair only if no expert tuning is required for alternative algorithms. Indeed, when using GloptiPoly the only assumption we make is that we know a ball containing the global optimizer. Besides this, our results are fully reproducible (Matlab files reproducing our examples are available upon request) and the SDP relaxations are solved with (black box) general-purpose semidefinite programming solvers.

6 Appendix on determinacy of measures on compact sets

Let $\mathbf{K} \subset \mathbb{R}^n$ be a compact set and let $C(\mathbf{K})$ be the Banach space of continuous functions on $\mathbf{K}$, equipped with the sup-norm. By Stone-Weierstrass, the polynomials are dense in $C(\mathbf{K})$. So let μ and ν be two finite Borel measures on $\mathbf{K}$ and assume that μ and ν have all same moment $\mathbf{y} = (y_\alpha)$, $\alpha \in \mathbb{N}^n$, i.e.,

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

Let $f \in C(\mathbf{K})$ be fixed arbitrary. As the polynomials are dense in $C(\mathbf{K})$ then $\int_{\mathbf{K}} f d\mu = \int_{\mathbf{K}} f d\nu$. And as f was arbitrary one concludes that this holds for every $f \in C(\mathbf{K})$. But this clearly implies that $\mu = \nu$. Hence every finite Borel measure on $\mathbf{K}$ is moment determinate.

7 Appendix on GloptiPoly scripts

7.1 Wilkinson-like rational function script

```
N = 20; mpol('x',N); % create variables
q = cell(N,1); % problem data
mu = cell(N,1); % measures
for i = 1:N, q{i} = i+x(i)^2; mu{i} = meas(x(i)); end
% model GMP
k = 0; % relaxation order
f = mass(mu{1}); % objective function
e = [mom(q{1}) == 1]; % moment contraints
for i = 2:N
 f = f + mass(mu{i});
 e = [e; mom(mmon(x(1),k)*q{1}) == mom(mmon(x(i),k)*q{i})];
end
% model SDP relaxation of GMP
P = msdp(max(f),e);
% solve SDP relaxation
[stat,obj] = msol(P)
```

In this Matlab script, the instructions `mpol`, `meas`, `mass`, `mom`, `mmon`, `msdp`, `max` and `msol` are GloptiPoly 3 commands, see the user's guide [12] for more information. Variable `f` is the objective function to be maximized. Since $p_i = 1$ for all $i = 1, \ldots, N$, it is the sum of masses of measures μ_i. Vector `e` stores the linear moment constraints and the instruction `mmon(x,k)` generates all monomials of variable `x` up to degree `k`. Finally, instruction `msdp` generates the SDP relaxation of the GMP, and `msol` solves the SDP problem with the default conic solver (SeDuMi 1.3 in our case).

7.2 Subsection 4.1.2 function script

```
mpol x1 x2
f1 = 1+x1+x1^2; g1 = 1+x1^2; f2 = 1+x2^2; g2 = 1+2*x2^2;
mu1 = meas(x1); mu2 = meas(x2);
bounds = [];
for k = 1:6
 P = msdp(min(mom(f1)+mom(f2)), ...
     mom(mmon(x1,2*(k-1))*g1) == mom(mmon(x2,2*(k-1))*g2), mom(g1) == 1);
 [stat, obj] = msol(P);
 bounds = [bounds; obj];
end
bounds
```

7.3 Subsection 4.2 function scripts

7.3.1 Script 1

```
mpol x1 x2 x3 x4
Pdense = msdp(min(x1*x2+x1*x3+x1*x4), ...
         x1^2+x2^2<=1,x1^2+x3^2<=2,x1^2+x4^2<=3,2);
[stat,obj] = msol(Pdense);
```

7.3.2 Script 2

```
mpol x1 3
mpol x2 x3 x4
mu(1) = meas([x1(1) x2]); % first measure on x1 and x2
mu(2) = meas([x1(2) x3]); % second measure on x1 and x3
mu(3) = meas([x1(3) x4]); % third measure on x1 and x4
f = mom(x1(1)*x2)+mom(x1(2)*x3)+mom(x1(3)*x4); % objective function
m1 = mom(mmon(x1(1),3)); % moments of first measure
m2 = mom(mmon(x1(2),3)); % moments of second measure
m3 = mom(mmon(x1(3),3)); % moments of third measure
K = [x1(1)^2+x2^2<=1, x1(2)^2+x3^2<=2, x1(3)^2+x4^2<=3]; % supports
Psparse = msdp(min(f),m1-m2==0,m2-m3==0,K,mass(mu)==1);
[stat,obj] = msol(Psparse);
```

Acknowledgments

We are grateful to Michel Devy, Jean-José Orteu, Tomáš Pajdla, Thierry Sentenac and Rekha Thomas for insightful discussions on applications of real algebraic geometry and SDP in computer vision, and to Josh Taylor for his feedback on the example of section 4.2.

References

[1] M. M. Ali, C. Khompatraporn, Z. B. Zabinsky. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Global Optim., 31(4):635-672, 2005.

[2] H. P. Benson. Global optimization algorithm for the nonlinear sum of ratios problem. J. Optimization Theory and Application, 112(1):1–29, 2002.

[3] H.P. Benson. Using concave envelopes to globally solve the nonlinear sum of ratios problems. J. Global Optimization, 22:343–364, 2002.

[4] S. J. Benson, Y. Ye. DSDP5 user guide — Software for semidefinite programming. Mathematics and Computer Science Division, Argonne National Laboratory, 2005.

[5] H. Bersini, M. Dorigo, S. Langerman, G. Seront, L. Gambardella. Results of the first international contest on evolutionary optimisation. IEEE Intl. Conf. Evolutionary Computation, Nagoya, Japan, 1996.

[6] B. Borchers. CSDP, a C library for semidefinite programming. Optimization Methods and Software, 11:613–623, 1999.

[7] R. E. Curto, L. A. Fialkow. Recursiveness, positivity, and truncated moment problems. Houston J. Math., 17:603–635, 1991.

[8] J. Czyzyk, M. Mesnier, J. Moré. The NEOS Server. IEEE J. Computational Science and Engineering, 5(3):68–75, 1998.

[9] R. W. Freund, F. Jarre. Solving the sum-of-ratios problem by an interior-point method. J. Global Optimization, 19(1):83–102, 2001.

[10] R. Hartley, F. Kahl, C. Olsson, Y. Seo. Verifying global minima for L2 minimization problems in multiple view geometry. International Journal of Computer Vision, 101(2):288–304, 2013.

[11] D. Henrion, J. B. Lasserre. GloptiPoly: global optimization over polynomials with Matlab and SeDuMi. ACM Trans. Math. Software, 29:165–194, 2003.

[12] D. Henrion, J. B. Lasserre, J. Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. Optim. Methods and Software, 24:761–779, 2009.

[13] D. Jibetean, E. de Klerk. Global optimization of rational functions: a semidefinite programming approach. Math. Programming, 106:93–109, 2006.

[14] F. Kahl, S. Agarwal, M. Chandraker, D. Kriegman, S. Belongie. Practical global optimization for multiview geometry. Int. J. Computer Vision, 79(3):271–284, 2008.

[15] F. Kahl, D. Henrion. Globally optimal estimates for geometric reconstruction problems. IEEE Intl. Conf. Computer Vision, Beijing, China, 2005.

[16] T. Kuno. A branch-and-bound algorithm for maximizing the sum of several linear ratios. J. Global Optimization, 22(1):155–174, 2002.

[17] J. B. Lasserre. Convergent SDP relaxations in polynomial optimization with sparsity. SIAM J. Optim., 17:822–843, 2006.

[18] J. B. Lasserre. A semidefinite programming approach to the generalized problem of moments. Math. Programming, 112:65–92, 2008.

[19] J. B. Lasserre. Moments, positive polynomials and their applications, Imperial College Press, London, 2010.

[20] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In Emerging Applications of Algebraic Geometry, 149:157–270, 2009.

[21] J. C. F. Pujol, R. Poli. A highly efficient function optimization with genetic programming. Genetic and Evolutionary Computation Conference, Seattle, Washington USA, 2004.

[22] M. Putinar. Positive polynomials on compact semi-algebraic sets, Ind. Univ. Math. J. 42:969984, 1993.

[23] S. Schaible, J. Shi. Fractional programming: the sum-of-ratios case. Optim. Methods Software, 18(2):219–229, 2003.

[24] M. Schweighofer. Optimization of polynomials on compact semialgebraic sets. SIAM Journal on Optimization, 17(3):805–825, 2005.

[25] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software, 12:625–653, 1999.

[26] M. Tawarmalani, N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. Mathematical Programming, 103(2):225–249,2005.

[27] K. C. Toh, M. J. Todd, R. H. Tutuncu. Solving semidefinite-quadratic-linear programs using SDPT3. Mathematical Programming serie B, 95:189–217, 2003.

[28] S. Waki, S. Kim, M. Kojima, M. Muramatsu. Sums of squares and semidefinite programming relaxations for polynomial optimization problemswith structured sparsity. SIAM J. Optim., 17:218–242, 2006.

[29] H. Waki, S. Kim, M. Kojima, M. Muramatsu, H. Sugimoto. SparsePOP: a sparse semidefinite programming relaxation of polynomial optimization problems. ACM Trans. Math. Software, 35:1–13, 2008.

[30] C. F. Wang, S. Y. Liu, P. P. Shen. Global optimization for sum of geometric fractional functions, Applied Mathematics and Computation, 216:2263–2270, 2010.

[31] W.-Y. Wu, R.-L. Sheu, S. Ilker Birbil. Solving the sum-of-ratios problem by a stochastic search algorithm. J. Glob. Optim., 42(1):91–109, 2008.

[32] M. Yamashita, K. Fujisawa, K. Nakata, M. Nakata, M. Fukuda, K. Kobayashi, K. Goto. A high-performance software package for semidefinite programs: SDPA 7. Research Report B-460 Dept. of Mathematical and Computing Science, Tokyo Institute of Technology, 2010.

[33] J. Löfberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.