

On Decompositions of Real Polynomials Using Mathematical Programming Methods

Janez Povh

Faculty of Information Studies Novo Mesto, Novo mesto, Slovenia
Institute of Mathematics, Physics and Mechanics Ljubljana, Ljubljana, Slovenia
E-mail: janez.povh@fis.unm.si

Received March 5, 2010; revised January 11, 2011; accepted January 15, 2011

Abstract

We present a procedure that gives us an SOS (sum of squares) decomposition of a given real polynomial in n variables, if there exists such decomposition. For the case of real polynomials in n non-commutative variables we extend this procedure to obtain a sum of hermitian squares (SOHS) decomposition whenever there exists any. This extended procedure is the main scientific contribution of the paper.

Keywords: Commutative Polynomial, Noncommutative Polynomial, Sum Of Squares, Semidefinite Programming, Newton Polytope

1. Introduction

Operations research, especially mathematical programming as subfield, very often makes progress by using results from other (“pure”) areas of mathematics, such as analysis, algebra, probability etc. Rarely happens the contrary, *i.e.* that the new methods, developed by mathematical programming community, inspires further research in pure mathematics.

One of the main breakthrough in operations research in last decades was development of theoretically and practically efficient interior-point methods for convex optimization problems. These methods were initially developed for linear programming (LP) problems and later extended to many other convex optimization problems. The first non-trivial class of optimization problems, to which the interior point methods were extended, was the class of linear programs over the cone of positive semidefinite matrices or over the second order cone (see Section 2 and [1]).

Once we became able to solve semidefinite programs efficiently (*i.e.* we can find ε -optimal solution in the time, which is a polynomial function of $\log \varepsilon$ and the size of the input data), the class of instances of the problems of this type and the methods to solve them (called *semidefinite programming*) became very important tool in optimization, control and also many other areas of applied mathematics and engineering, see e.g. [1].

Computing global minimum of a given real poly-

nomial over the set, defined by polynomial inequalities (we call such set a semi-algebraic set) is an NP-hard problem, since it includes linear binary problems. Several authors, in particular Lasserre [2-4], Parillo [5], Parrilo and Sturmfels [6] and Schweighofer [7,8] have shown how to solve such a problem approximately by a sequence of semidefinite programs.

The main results needed to construct such a sequence are the fact that if a given polynomial is a sum of squares (SOS), then it is non-negative, and the dual theory of moments [2]. Precise and comprehensive overview of the results about topic has been done by M. Laurent in [9].

Checking whether given polynomial is non-negative is an NP-hard problem, but checking if given polynomial is SOS can be done efficiently in theory and practise by semidefinite programming. This topic has been well exploited in recent years and currently there are available software packages to detect SOS polynomials and to do polynomial optimization. Readers interested in solving sums of squares problems for commuting polynomials are referred to one of the great existing packages SOSTOOLS [10,11], GloptiPoly [12], YALMIP [13,14], and SparsePOP [15].

Comparing to commutative polynomials is the problem of writing a non-commutative polynomial as a sum of hermitian squares (SOHS) much less exploited—theoretically and practically. However, several results show that this area is interesting and important.

Helton [16] proved that real given polynomial in NC

variables is SOHS if and only if it yields a positive semi-definite matrix (PSD) after substituting the variable by symmetric real matrices of the same size. For a beautiful exposition, we refer the reader to [17].

Together with coworkers Helton pursued line of research of NC polynomials further, studied positivity and convexity of NC polynomials and gave applications to control theory, optimization, systems engineering, etc.; see [18] for a nice survey of these beginnings of *free semialgebraic geometry*. The first author in [19] connected sums of hermitian squares of NC polynomials to an old open problem of Connes on von Neumann algebras, and, somewhat related, found applications to mathematical physics [20]. Many of these results were obtained with the aid of computer programs written in an ad-hoc manner.

Despite the fast rise of free semialgebraic geometry, it seems that [21] is the first publication about theoretical algorithm and publicly available software for computing (or determining existence) of sums of hermitian squares (SOHS) decompositions of NC polynomials.

In this paper we illustrate the procedures mentioned above on a number of well-chosen examples.

1.1. Notation

In the paper we use standard notation from optimization and algebra. By $X \succeq 0$ we denote that X is positive semidefinite (i.e. X is symmetric and has only non-negative eigenvalues). The scalar product of two matrices X and Y of the same size is $\langle X, Y \rangle = \text{trace}(X^T Y) = \sum_{i,j} x_{ij} y_{i,j}$. By I we denote the identity matrix.

A set of real polynomials in n (commutative) variables (algebraists call this set *algebra* of real polynomials) is denoted by $\mathbb{R}[\bar{x}] = \mathbb{R}[x_1, \dots, x_n]$. Elements of $\mathbb{R}[\bar{x}]$ are therefore polynomials, which can be written as follows

$$p(\bar{x}) = \sum_i c_i \bar{x}^{\alpha_i} = \sum_i c_i x_1^{\alpha_i(1)} x_2^{\alpha_i(2)} \dots x_n^{\alpha_i(n)},$$

where α_i are vectors from \mathbb{N}^n , which define the exponents of monomials. The degree of monomial x^α is $\text{deg}(x^\alpha) = \sum_i \alpha(i)$ and the degree of polynomial $p(\bar{x}) = \sum_i c_i x^{\alpha_i}$ is $\text{deg}(p) = \max_j \sum_i \alpha_i(j)$. If all monomials in p have the same degree d , we say that p is d -form.

Note that the coefficient c_i and the exponent vectors α_i completely determine the polynomial.

Example 1. Polynomial $p(x, y) = 10x^5 y^3 - 2x^2 y^7$ has vectors of exponents $\alpha_1 = (5, 3)$ and $\alpha_2 = (2, 7)$. We have $\text{deg}(p) = 9$.

The set (free algebra) of *non-commutative* polynomials in variables $\bar{x} = (x_1, x_2, \dots, x_n)$ is denoted by $\mathbb{R}\langle \bar{x} \rangle$ and consists of linear combinations of all finite words

with letters x_1, \dots, x_n . We endow $\mathbb{R}\langle \bar{x} \rangle$ with the involution $*$, which reverses the order of the letters in any word, i.e. it holds $(p+q)^* = p^*+q^*$ and $(pq)^* = q^* p^*$. The degree of polynomial is length of the longest word.

Example 2. Let $p(x) = x^2 y - xy^2 x^2 + 1 \in \mathbb{R}\langle x, y \rangle$.

We have $p^* = yx^2 - x^2 y^2 x + 1$ and $\text{deg}(p) = 5$.

Remark 1. In the non-commutative case we can not express monomials simply by the vectors of exponents, since monomials $x^2 y$ and $y x^2$, which are in commutative case equal with the same exponent $(2, 1)$, are no more equal in the non-commutative case. Instead of this we keep working with monomials, i.e. with the words.

2. Semidefinite Programming

Semidefinite programming consists of problems, where we are interested for the optimum of a linear function subject to linear constraints and additional constraint that all variables are taken from positive semidefinite matrix. More precisely, given symmetric matrices C, A_1, \dots, A_m and vector $b \in \mathbb{R}^m$, we formulate semidefinite program in standard primal form (in the sequel we refer to problems of this type by PSDP) as follows:

$$\begin{aligned} \min \quad & \langle C, X \rangle && (PSDP) \\ \text{s.t.} \quad & \langle A_i, X \rangle = b_i, \quad i=1, \dots, m, \\ & X \succeq 0. \end{aligned}$$

The conic dual problem to PSDP is the semidefinite program in the standard dual form (DSDP)

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & \sum_i y_i A_i + Z = C, \quad (DSDP) \\ & y \in \mathbb{R}^m, Z \succeq 0. \end{aligned}$$

As we already mentioned, the importance of semidefinite programming was spurred by development of efficient methods, which can find ε -optimal solution in a polynomial time of n, m and $\log \varepsilon$, where n is the order of matrix variables Z and X . There exist several freeware package, which also in practice find such solutions. If the problem is of the medium size (i.e. $n \leq 1000$ and $m \leq 10.000$), these packages are based on the interior point methods, while packages for larger semidefinite programs use some variant of the first order methods (see web page [22] with a comprehensive list of state-of-the-art SDP solvers and also [23]).

3. SOS Decompositions of Commutative Polynomials by SDP

SOS decomposition appears naturally when we are interested in finding the global optimum of a given poly-

nomial. Indeed, the problem

$$opt_p = \min \{ p(\bar{x}) : \bar{x} \in \mathbb{R}^n \} \tag{1}$$

is equivalent to

$$\max \{ \lambda : p(\bar{x}) - \lambda \geq 0 \}.$$

Both problems are in general very difficult (*i.e.* NP-hard), therefore we are forced to relax the problems in order to obtain a tractable one. One of the possibilities is using the SOS decomposition.

Definition 1. Polynomial $p \in \mathbb{R}[\bar{x}]$ has a sum of squares decomposition if there exist polynomials q_1, \dots, q_k such that $p = \sum_i q_i^2$.

If the polynomial p can be written as a sum of squares, then clearly we have $p \geq 0$, while the converse is in general not true. This leads to the following lower bound for the problem (1):

$$opt_p \geq \max \{ \lambda : p(\bar{x}) - \lambda \text{ is SOS} \}.$$

The inequality from above might be strict, as also follows from the following example.

Example 3. Polynomial

$$M(x_1, x_2, x_3) = x_1^4 x_2^2 + x_1^2 x_2^4 + x_3^6 - 3x_1^2 x_2^2 x_3^2$$

is well-known as Motzkin form and is defined by four coefficients and four vectors:

$$c_1 = 1, \alpha_1 = (4, 2, 0), c_2 = 1, \alpha_2 = (2, 4, 0), c_3 = 1, \alpha_3 = (0, 0, 6)$$

in $c_4 = -3, \alpha_4 = (2, 2, 2)$ It has been shown (see [24, 25]) that $M(x_1, x_2, x_3) \geq 0$ while M has no SOS decomposition.

There exist only few sets of polynomials where it holds that a polynomial from this set is non-negative if and only if it has SOS decomposition. This is true for all polynomials in two variables, for all 2-forms and for 4-forms in 3 variables. Obviously the Motzkin form from above is not in any of these sets.

Testing whether a given polynomial has SOS decomposition can be done efficiently by semidefinite programming, as follows from the following theorem. For the proof see e.g. a proof of non-commutative version [21, Prop. 2.1].

Theorem 1. Polynomial $p \in \mathbb{R}[\bar{x}]$ has SOS decomposition if and only if there exists a positive semidefinite matrix Q such that

$$p(\bar{x}) = U^T Q U,$$

where U is the column containing all monomials of degree $\leq \text{deg}(p)/2$.

We demonstrate this theorem by the following example.

Example 4. Let us consider

$$p(x_1, x_2) = 2x_1^4 + 2x_1^3 x_2 - x_1^2 x_2^2 + 5x_2^4.$$

Since p is 4-form, it follows from above that p has SOS decomposition if and only if it is non-negative. Moreover, in the SOS decomposition we can have only monomials $x_1^2, x_2^2, x_1 x_2$.

Polynomial p has therefore the SOS decomposition if there exists positive semidefinite matrix Q such that $p(x_1, x_2) = U^T Q U$, for $U = (x_1^2, x_2^2, x_1 x_2)^T$. For the matrix Q we have linear constraints that its components must coincide with the coefficients of p . We have in p monomial x_1^4 with coefficient 2, therefore it must hold $q_{1,1} = 2$ and similarly $q_{2,2} = 5$. Monomial $x_1^3 x_2$ has coefficient 2 in and can be obtained as $x_1^2 \cdot x_1 x_2$, hence we have also constraint $q_{1,3} + q_{3,1} = 2$ or equivalently $q_{1,3} = q_{3,1} = 1$. Monomial $x_1^2 x_2^2$ can be obtained as a product of x_1^2 in x_2^2 or as a square of $x_1 x_2$, therefore $q_{1,2} + q_{2,1} + q_{3,3} = -1$. Monomial $x_1 x_2^3$ does not appear p , hence $q_{2,3} + q_{3,2} = 0$ and since Q is symmetric: $q_{2,3} = q_{3,2} = 0$. If we put things together, we are looking for a positive semidefinite matrix Q with components satisfying the constraints above. Note that the objective function is not important here, *i.e.* we have the *semidefinite feasibility problem*. We can solve it by heart: by setting $q_{3,3} = 5$ we obtain $q_{1,2} = q_{2,1} = -3$. The matrix is now completely defined:

$$Q = \begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix}^T \cdot \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

We have the following SOS decomposition:

$$p(x_1, x_2) = \frac{1}{2} \left((2x_1^2 - 3x_2^2 + x_1 x_2)^2 + (x_2^2 + 3x_1 x_2)^2 \right).$$

Remark 2. Even though we have an SDP feasibility problem, we usually use the following objective function $\text{trace}(Q)$, since it is widely accepted as an efficient heuristics to obtain the feasible solution with lowest rank. This is important since the rank of Q is exactly the number of factors in SOS decomposition.

In general we must include in the vector U all monomials of degree d , if the polynomial is $2d$ -form. We have $\binom{n+d-1}{d}$ of such monomials. If the polynomial has degree $2d$, but is not $2d$ -form, the vector U contains all possible monomials of degree $\leq d$ - there are $\binom{n+d}{d}$ many of them. We can find examples, where we indeed need all these monomials. If $n = 4$ and $d = 10$, we obtain $\binom{n+d-1}{d} = 286$ and $\binom{n+d}{d} = 1001$, hence the resulting SDPs are already on the boundary of the set of instances, solvable by interior point methods.

Nevertheless, very often, especially if the polynomial

is sparse (has only few monomials) it is possible to considerable decrease the number of monomials in U . We can use a result, first formulated in [26], that characterizes the monomials that can appear in a sum of squares representation. Define the Newton polytope \mathcal{N}_p of given polynomial p of degree $2d$ as the integer lattice points in the convex hull of the degrees α , which appear in p . Then, it can be shown that the only monomials x^β that can appear in a sum of squares representation are those such that 2β is in the \mathcal{N}_p (or equivalently $\beta \in \frac{1}{2}\mathcal{N}_p$).

Example 5. If $\frac{2}{p}$ is from Example 4, we have

$$\mathcal{N}_p = \mathbb{Z}^2 \cap \text{CONV}\{(4,0),(3,1),(2,2),(0,4)\}$$

and

$$\frac{1}{2}\mathcal{N} = \{(2,0),(1,1),(0,2)\}$$

hence it defines exactly the monomials which appear in the SOS decomposition.

The package SO Stools and some other packages for SOS decompositions are essentially based on the Newton polytope algorithm.

4. SOHS Decompositions for Non-Commutative Polynomials

A non-commutative polynomial $p \in \mathbb{R}\langle \bar{x} \rangle$ has a sum-of-Hermitian-squares (SOHS) decomposition if there exist polynomials q_1, \dots, q_k such that $p = \sum_i q_i^* q_i$. One can prove a result, similar to Theorem 1: $p \in \mathbb{R}\langle \bar{x} \rangle$ of degree $2d$ has SOHS decomposition if and only if

$$p = U^* Q U, \tag{2}$$

for some $Q \succeq 0$, where U is vector of all monomials of degree d .

If we consider all possible monomials of degree d , then vector U has $\binom{n^{d+1}-1}{d-1}$ components. This is much larger comparing to the commutative case. Nevertheless, we have also many criteria which tell us when we can leave out some of the monomials from U . We demonstrate the procedure to find SOHS decomposition by the following example.

Example 6. Let

$$p(x_1, x_2) = 1 + 3x_1^2 - x_1x_2 - 6x_1x_2x_1 + 3x_1x_2^2x_1 - x_2x_1 + x_2x_1^2x_2$$

Since p has degree 4, there are $2^3 - 1 = 7$ monomials which might appear in the vector U from the SOHS decomposition:

$$U = (1, x_1, x_2, x_1x_2, x_2x_1, x_1^2, x_2^2)^T.$$

Due to the structure of p we can cross out some com-

ponents of U : for any monomial from U of (maximum) degree 2 it must hold that its hermitian square must be in p . If this is not the case for some monomial, we can eliminate it from U .

Therefore we can cross out x_1^2 in x_2^2 from U to obtain

$$U = (1, x_1, x_2, x_1x_2, x_2x_1)^T.$$

We have

$$p(x_1, x_2) = U^* Q U \tag{3}$$

if and only if Q satisfies equations:

$$\begin{aligned} q_{1,1} &= 1 \\ q_{2,2} &= 3 \\ q_{2,3} + q_{3,2} + q_{1,4} + q_{4,1} + q_{1,5} + q_{5,1} &= -2 \\ q_{2,5} + q_{5,2} &= -6 \\ q_{5,5} &= 3 \\ q_{4,4} &= 1. \end{aligned}$$

Beside these constraints we get also a bunch of homogenous equations: since x_2^2 does not appear in p , Q must satisfy $q_{3,3} = 0$. We also miss $x_1^2x_2$ and $x_2x_1^2$ in p , hence $q_{2,4} + q_{3,5} + q_{4,2} + q_{5,3} = 0$. We obtain 13 equations in total, hence we have to solve SDP of order 5 with 13 linear constraints.

By using objective $\text{trace}(Q)$ (note Remark 2) our SDP solver (Sedumi [27]) gives the following optimal solution

$$Q = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & 3 & 0 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -3 & 0 & 0 & 3 \end{bmatrix} = P^T P$$

where

$$P = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 \\ 0 & \sqrt{3} & 0 & 0 & -\sqrt{3} \end{bmatrix}.$$

Therefore we obtain

$$p(x_1, x_2) = (1 - x_1x_2)^*(1 - x_1x_2) + 3(x_1 - x_2x_1)^*(x_1 - x_2x_1).$$

Note that x_2 does not appear in the SOHS decomposition, therefore we could eliminate it from U .

Previous example clearly shows the main steps of the general algorithm to obtain the SOHS decomposition, presented in **Figure 1**.

We can obtain in general case much larger SDP in the non-commutative case, but we have also much more possibilities to exploit the fact that the polynomials are non-commutative in order to reduce the size of the SDP. In

INPUT: $p \in \mathbb{R}\langle \bar{x} \rangle$

- 1) Construct the vector of possible monomials U .
- 2) Construct linear equations on matrix Q .
- 3) Solve SDP in variable Q .
- 4) IF the SDP is infeasible, then p has no SOHS decomposition.

RETURN.

- 5) ELSE compute p such that $Q = P^T P$. Let q_i be the i th component of PX .

OUTPUT: SOHS decomposition $p = \sum_i q_i^* q_i$.

Figure 1. Algorithm to obtain the SOHS decomposition.

particular, we can execute the following:

- If monomial m appears in the SOHS decomposition and must therefore appear in U , then there should exist a symmetric monomial in p which coincides with m in the $|m|$ rightmost variables (letters).
- At least one of the monomials of highest and of lowest degree must be symmetric.
- p must be symmetric, *i.e.* $p^* = p$.
- For every variable x_i the monomial, where x_i has highest/lowest degree, must be symmetric.

The first observation is very important: it leads to the so-called Newton chip method [21], where we obtain the vector U simply by taking all possible right chips (tails) of all possible symmetric monomials in p , which have length between half of the minimum and half of the maximum degree in p , see also paper [21].

We have also written a Matlab based software package NCsostools, which is a non-commutative version of the Sostools package. It includes an implementation of the algorithm from above and also implementations of most important operations from $\mathbb{R}\langle \bar{x} \rangle$. It is freely available from <http://ncsostools.fis.unm.si>, see also the detailed description of commands [28]. However, since all constraints in the resulting SDP are always orthogonal, we see a strong potential in the boundary point method [23], which performs very good on such SDP problems, especially if we have large scale SDP.

5. Conclusions

In this paper we demonstrated the power of mathematical programming in other, more pure areas of mathematics. Semidefinite programming turned out to be a very strong tool in approximating optimal values of polynomials, since it gives SOS or SOHS decompositions of real polynomials, when they exist. The next step of the research will be publishing the NCsostools package, which will contain implementation of the algorithm for finding the SOHS decompositions and also implementations of

most important operations in $\mathbb{R}\langle \bar{x} \rangle$.

A very challenging task for future research is a procedure to extract an exact rational solution of the SDP from the numerical solution Q , obtained by SDP solver. It is very important since in practice SDP solvers return only approximately feasible solutions, which are sufficient for most purposes. But if we want to have correct proof for SOS or SOHS decomposition, we need an exact rational solution. This is in general very difficult problem (NP-hard) and not much research has been done in this direction.

6. Acknowledgements

We gratefully acknowledge financial supported by the Slovenian Research Agency (project No. 1000-08-210518).

7. References

- [1] H. Wolkowicz, R. Saigal and L. Vandenberghe, "Hand-Book of Semidefinite Programming," Kluwer, Academic Publishers, Boston, 2000.
- [2] J. B. Lasserre, "Global Optimization with Polynomials and the Problem of Moments," *SIAM Journal on Optimization*, Vol. 11, No. 3, January 2000, pp. 796-817. doi:10.1137/S1052623400366802
- [3] J. -B. Lasserre, "A Sum of Squares Approximation of Nonnegative Polynomials," *SIAM Journal on Optimization*, Vol. 16, No. 3, 2006, pp. 751-765. doi:10.1137/04061413X
- [4] D. Henrion and J. -B. Lasserre, "GloptiPoly: Global Optimization over Polynomials with Matlab and SeDuMi," *ACM Transactions on Mathematical Software*, Vol. 29, No. 2, 2003, pp. 165-194. doi:10.1145/779359.779363
- [5] P. A. Parrilo, "Semidefinite Programming Relaxations for Semialgebraic Problems," *Mathematical Programming*, Vol. 96, No. 2, Series B, 2003, pp. 293-320.
- [6] P. A. Parrilo and B. Sturmfels, "Minimizing Polynomial Functions," In *Algorithmic and Quantitative Real Algebraic Geometry* (Piscataway, NJ, 2001), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, *American Mathematical Society*, Providence, RI, Vol. 60, 2003, pp. 83-99.
- [7] M. Schweighofer, "Optimization of Polynomials on Compact Semialgebraic Sets," *SIAM Journal on Optimization*, Vol. 15, No. 3, 2005, pp. 805-825. doi:10.1137/S1052623403431779
- [8] M. Schweighofer, "Global Optimization of Polynomials Using Gradient Tentacles and Sums of Squares," *SIAM Journal on Optimization*, Vol. 17, No. 3, 2006, pp. 920-942. doi:10.1137/050647098
- [9] M. Laurent, "Sums of Squares, Moment Matrices and Optimization over Polynomials," In *Emerging Applications of Algebraic Geometry*, *IMA Volumes in Mathematics and Its Applications*, Springer, New York, Vol.

- 149, 2009, pp. 157-270.
- [10] SOSTools, 2011. <http://www.cds.caltech.edu/sostools/>.
- [11] S. Prajna, A. Papachristodoulou, P. Seiler and P. A. Parrilo, "SOSTOOLS and Its Control Applications," *Positive Polynomials in Control*, Lecture Notes in Control and Information and Science, Springer, Berlin, Vol. 312, 2005, pp. 273-292.
- [12] D. Henrion, J.-B. Lasserre and J. Löfberg, "GloptiPoly 3: Moments, Optimization and Semidefinite Programming," 2011. <http://www.laas.fr/~henrion/software/gloptipoly3/>
- [13] YALMIP, 1 March 2009. <http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php>
- [14] J. Löfberg, "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," *Proceedings of the CACSD Conference*, Taipei, 2004. <http://control.ee.ethz.ch/~joloef/yalmip.php>.
- [15] H. Waki, S. Kim, M. Kojima and M. Muramatsu, "Sums of Squares and Semidefinite Program Relaxations for Polynomial Optimization Problems with Structured Sparsity," *SIAM Journal on Optimization*, Vol. 17, No. 1, 2006, pp. 218-242. doi:10.1137/050623802
- [16] J. W. Helton, "'Positive' Noncommutative Polynomials are Sums of Squares," *Annals of Mathematics*, Vol. 156, No. 2, 2002, pp. 675-694. doi:10.2307/3597203
- [17] S. McCullough and M. Putinar, "Noncommutative Sums of Squares," *Pacific Journal of Mathematics*, Vol. 218, No. 1, 2005, pp. 167-171. doi:10.2140/pjm.2005.218.167
- [18] M. C. de Oliveira, J. W. Helton, S. McCullough and M. Putinar, "Engineering Systems and Free Semi-Algebraic Geometry," *Emerging Applications of Algebraic Geometry, IMA Volumes in Mathematics and Its Applications*, Springer, Vol. 149, 2008, pp. 17-62.
- [19] I. Klep and M. Schweighofer, "Connes' Embedding Conjecture and Sums of Hermitian Squares," *Advances in Mathematics*, Vol. 217, No. 4, 2008, pp. 1816-1837. doi:10.1016/j.aim.2007.09.016
- [20] I. Klep and M. Schweighofer, "Sums of Hermitian Squares and the BMV Conjecture," *Journal of Statistical Physics*, Vol. 133, No. 4, 2008, pp. 739-760. doi:10.1007/s10955-008-9632-x
- [21] I. Klep and J. Povh, "Semidefinite Programming and Sums of Hermitian Squares of Noncommutative Polynomials," *Journal of Pure and Applied Algebra*, Vol. 214, No. 6, 2010, pp. 740-749. doi:10.1016/j.jpaa.2009.07.003
- [22] H. Mittelman, 2011. <http://plato.asu.edu/sub/pns.html>.
- [23] J. Povh, F. Rendl and A. Wiegele, "A Boundary Point Method to Solve Semidefinite Programs," *Computing*, Vol. 78, No. 3, 2006, pp. 277-286. doi:10.1007/s00607-006-0182-2
- [24] P. Parrilo, "Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization," PhD Thesis, California Institute of Technology, California, 2000.
- [25] B. Reznick, "Some Concrete Aspects of Hilbert's 17th Problem," In *Real Algebraic Geometry and Ordered Structures* (Baton Rouge, LA, 1996), *Contemporary Mathematics, American Mathematical Society*, Providence, RI, Vol. 253, 2000, pp. 251-272.
- [26] B. Reznick, "Extremal PSD Forms with Few Terms," *Duke Mathematical Journal*, Vol. 45, No. 2, 1978, pp. 363-374. doi:10.1215/S0012-7094-78-04519-2
- [27] SeDuMi, 29 June 2009. <http://sedumi.ie.lehigh.edu/>
- [28] K. Cafuta, I. Klep and J. Povh, "NCSOSTools: A Computer Algebra System for Symbolic and Numerical Computation with Noncommutative Polynomials," 2011. <http://ncsostools.fis.unm.si>