Solving Global Optimization Problems over Polynomials with GloptiPoly 2.1

Didier Henrion[1,2] and Jean-Bernard Lasserre[1]

[1] Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique, 7 Avenue du Colonel Roche, 31 077 Toulouse, cedex 4, France
{henrion,lasserre}@laas.fr

[2] Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 4, 182 08 Praha, Czech Republic
henrion@utia.cas.cz

Abstract. GloptiPoly is a Matlab/SeDuMi add-on to build and solve convex linear matrix inequality relaxations of the (generally non-convex) global optimization problem of minimizing a multivariable polynomial function subject to polynomial inequality, equality or integer constraints. It generates a series of lower bounds monotonically converging to the global optimum. Global optimality is detected and isolated optimal solutions are extracted automatically. In this paper we first briefly describe the theoretical background underlying the relaxations. Following a small illustrative example of the use of GloptiPoly, we then evaluate its performance on benchmark test examples from global optimization, combinatorial optimization and polynomial systems of equations.

1 Introduction

GloptiPoly is a Matlab[3] freeware that builds and solves convex linear matrix inequality (LMI, see [VB96]) relaxations of (generally non-convex) global optimization problems with multivariable real-valued polynomial objective function and constraints. The software solves a series of convex relaxations of increasing size, whose optima are guaranteed to converge monotonically to the global optimum of the original non-convex optimization problem.

GloptiPoly solves LMI relaxations with the help of the semidefinite programming (SDP) solver SeDuMi [SDM99], taking full advantage of sparsity and special problem structure. Optionally, a user-friendly interface called DefiPoly, based on Matlab Symbolic Math Toolbox, can be used jointly with GloptiPoly to define the optimization problems symbolically with a Maple-like syntax.

GloptiPoly is aimed at small- and medium-scale problems. Numerical experiments illustrate that for most of the problem instances available in the literature, the global optimum is reached exactly with LMI relaxations of medium size, at a relatively low computational cost.

[3] Matlab is a trademark of The MathWorks, Inc.

GloptiPoly requires Matlab version 5.3 or higher [Mat01], together with the freeware solver SeDuMi version 1.05 [SDM99]. For installation instructions and a comprehensive user's guide, see

`www.laas.fr/~henrion/software/gloptipoly`

2 Theoretical background

GloptiPoly is based on the theory of positive polynomials and moments described in [Las01,Las02] and briefly summarized in the sequel.

2.1 Introduction

Consider the general nonlinear optimization problem

$$\mathbb{P} \to p^* := \min_{x \in \mathbb{R}^n} \{g_0(x) \,|\, g_k(x) \geq 0,\ k = 1, \ldots m\} \tag{1}$$

where all the $g_k(x) : \mathbb{R}^n \to \mathbb{R}$ are real-valued polynomials of $\mathbb{R}[x_1, \ldots, x_n]$. Equality constraints are allowed via two opposite inequalities, so that (1) describes all optimization problems that involve polynomials. In particular, it encompasses non-convex quadratic problems as well as discrete optimization problems (e.g. 0-1 nonlinear programming problems).

The idea behind the methodology of GloptiPoly is to build up a sequence of convex semidefinite relaxations of $\mathbb{P}$ of increasing size and whose sequence of optimal values converges to the global optimal value $p^* = \inf \mathbb{P}$.

The original idea can be traced back to the pioneering Reformulation Linearization Technique (RLT) of [SA90,SA99] where additional redundant constraints (products of the original ones) are introduced and linearized in a higher space (lifting) by introducing additional variables (e.g. $x_i x_j = y_{ij}$) so as to obtain a LP-relaxation. Convergence was proved for 0-1 nonlinear programs. Later, Shor [Sho87,Sho98] also proposed a lifting procedure to reduce any polynomial programming problem to a quadratic one and then use a semidefinite relaxation to obtain a lower bound of p^*, see also the more recent work [Nes00]. Then, the striking certified good approximation of Goemans and Williamson for the MAX-CUT problem [GW95], obtained from a simple SDP (or LMI) relaxation definitely excited the curiosity of researchers for SDP relaxations. However, excepted for the LP-relaxations of Sherali and Adams in 0-1 problems, no proof of convergence was provided.

The proof of convergence of the LMI relaxations defined in [Las01,Las02] and used in GloptiPoly is based on recent results of real algebraic geometry concerning the representation of polynomials, strictly positive on a semi-algebraic set; see also [Par00] for a related approach. It turns out that the primal and dual LMI relaxations of GloptiPoly match both sides of the dual theories of *moments* and *positive polynomials*.

Indeed, while the primal relaxations aim at founding the moments of a probability measure with mass concentrated on some global minimizers of $\mathbb{P}$, the dual relaxations aim at representing the polynomial $g_0(x) - p^*$, nonnegative on the (semi-algebraic) feasible set $\mathbb{K}$ of $\mathbb{P}$, as a linear combination of the g_i's with weights being polynomials that are sums of squares, as in Putinar's representation of polynomials, strictly positive on a semi-algebraic set [Put93].

In brief, the primal LMI relaxations $\{\mathbb{Q}_i\}$ of $\mathbb{P}$ are relaxations of the problem (equivalent to $\mathbb{P}$)

$$p^* \;=\; \min_{\mu}\{\int g_0 \, d\mu \mid \mu(\mathbb{K}) = 1\},$$

where the minimum is taken over all the probability measures on the feasible set $\mathbb{K}$ of $\mathbb{P}$, whereas the dual relaxations $\{\mathbb{Q}_i^*\}$ solve

$$\max_{\rho_i, \{q_k\}} \{\rho_i \mid g_0(x) - \rho_i = q_0 + \sum_{k=1}^{m} g_k(x) q_k(x)\}, \tag{2}$$

where the unknowns $\{q_k\}$ are polynomials, all sums of squares, and with degree at most $2i$. For a brief account of these two dual points of view, the interested reader is referred to [Las01,Las02] and the references therein.

The increasing size of the relaxations reflects the effort in the degree $2i$ needed in (2) for ρ_i to be as closed as desired to p^* (and often to be exactly equal to p^*).

2.2 Brief description of the methodology

Notation and definitions: Given any two real-valued symmetric matrices A, B let $\langle A, B\rangle$ denote the usual scalar product trace(AB) and let $A \succeq B$ (resp. $A \succ B$) stand for $A - B$ positive semidefinite (resp. $A - B$ positive definite). Let

$$1, x_1, x_2, \ldots x_n, x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, x_2x_3, \ldots, x_n^2, \ldots, x_1^r, \ldots, x_n^r, \tag{3}$$

be a basis for the space $\mathcal{A}_r$ of real-valued polynomials of degree at most r, and let $s(r)$ be its dimension. Therefore, a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ of degree r is written

$$p(x) \;=\; \sum_{\alpha} p_\alpha x^\alpha, \qquad x \in \mathbb{R}^n,$$

where

$$x^\alpha \;=\; x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}, \quad \text{with} \; \sum_{i=1}^{n} \alpha_i = k,$$

is a monomial of degree k with coefficient p_α. Let $p = \{p_\alpha\} \in \mathbb{R}^{s(r)}$ be the vector of coefficients of the polynomial $p(x)$ in the basis (3).

Given an $s(2r)$-sequence $(1, y_1, \ldots,)$, let $M_r(y)$ be the moment matrix of dimension $s(r)$ with rows and columns indexed by (3). For instance, to fix ideas, consider the 2-dimensional case. The moment matrix $M_r(y)$ is the block matrix $\{M_{i,j}(y)\}_{0 \le i,j \le r}$ defined by

$$M_{i,j}(y) \;=\; \begin{bmatrix} y_{i+j,0} & y_{i+j-1,1} & \cdots & y_{i,j} \\ y_{i+j-1,1} & y_{i+j-2,2} & \cdots & y_{i-1,j+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_{j,i} & y_{i+j-1,1} & \cdots & y_{0,i+j} \end{bmatrix}.$$

Thus, with $n = 2$ and $r = 2$, one obtains

$$M_2(y) \; = \; \left[\begin{array}{c|cc|ccc} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{0,2} \\ \hline y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ \hline y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{array}\right]$$

Another, more intuitive way of constructing $M_r(y)$ is as follows. If $M_r(y)(1,i) = y_\alpha$ and $M_r(y)(j,1) = y_\beta$, then $M_r(y)(i,j) = y_{\alpha+\beta}$, with $\alpha + \beta = (\alpha_1 + \beta_1, \cdots, \alpha_n + \beta_n)$. This defines a bilinear form $\langle ., . \rangle_y$ on $\mathcal{A}_r$, by $\langle q(x), v(x) \rangle_y := \langle q, M_r(y)v \rangle, \quad q(x), v(x) \in \mathcal{A}_r$, and if y is a sequence of moments of some measure μ_y, then

$$\langle q, M_r(y)q \rangle \; = \; \int q(x)^2 \, \mu_y(dx) \; \geq \; 0, \tag{4}$$

so that $M_r(y) \succeq 0$.

If the entry (i,j) of the matrix $M_r(y)$ is y_β, let $\beta(i,j)$ denote the subscript β of y_β. Next, given a polynomial $\theta(x) : \mathbb{R}^n \to \mathbb{R}$ with coefficient vector θ, we define the matrix $M_r(\theta y)$ by

$$M_r(\theta y)(i,j) \; = \; \sum_\alpha \theta_\alpha y_{\{\beta(i,j)+\alpha\}}. \tag{5}$$

For instance, with

$$M_1(y) = \left[\begin{array}{c|cc} 1 & y_{10} & y_{01} \\ \hline y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{array}\right] \quad \text{and } x \; \mapsto \; \theta(x) \; := \; a - x_1^2 - x_2^2,$$

we obtain

$$M_1(\theta y) = \left[\begin{array}{c|cc} a - y_{20} - y_{02} & ay_{10} - y_{30} - y_{12} & ay_{01} - y_{21} - y_{03} \\ \hline ay_{10} - y_{30} - y_{12} & ay_{20} - y_{40} - y_{22} & ay_{11} - y_{31} - y_{13} \\ ay_{01} - y_{21} - y_{03} & ay_{11} - y_{31} - y_{13} & ay_{02} - y_{22} - y_{04} \end{array}\right].$$

In a manner similar to what we have in (4), if y is a sequence of moments of some measure μ_y, then

$$\langle q, M_r(\theta y)q \rangle \; = \; \int \theta(x) q(x)^2 \, \mu_y(dx),$$

for every polynomial $q(x) : \mathbb{R}^n \to \mathbb{R}$ with coefficient vector $q \in \mathbb{R}^{s(r)}$. Therefore, $M_r(\theta y) \succeq 0$ whenever μ_y has its support contained in the set $\{\theta(x) \geq 0\}$. The matrix $M_r(\theta y)$ is called a *localizing* matrix.

The $\mathbb{K}$-moment problem identifies those sequences y that are moment-sequences of a measure with support contained in the semi-algebraic set $\mathbb{K}$. In duality with the theory of moments is the theory of representation of positive polynomials, which dates back to Hilbert's 17th problem. This fact will be reflected in the semidefinite relaxations proposed later.

LMI relaxations: Let $\mathbb{P}$ be the problem defined in (1) and let

$$\mathbb{K} := \{x \in \mathbb{R}^n \,|\, g_k(x) \geq 0,\ k = 1, \ldots, m\} \tag{6}$$

be the feasible set associated with $\mathbb{P}$.

Depending on its parity, let degree$(g_k) = 2v_k - 1$ or $2v_k$, for all $k = 0, 1, \ldots, m$. When needed below, for $i \geq \max_k v_k$, the vectors $g_k \in \mathbb{R}^{s(2v_k)}$ are extended to vectors of $\mathbb{R}^{s(2i)}$ by completing with zeros. As we minimize $g_0(x)$ we may and will assume that its constant term is zero, that is, $g_0(0) = 0$.

For $i \geq \max_{k\in\{0,m\}} v_k$, consider the following family $\{\mathbb{Q}_i\}$ of convex positive semidefinite programs, or LMI relaxations of $\mathbb{P}$:

$$\mathbb{Q}_i \begin{cases} \min\limits_y \displaystyle\sum_\alpha (g_0)_\alpha y_\alpha \\ \qquad\qquad M_i(y) \succeq 0 \\ \qquad M_{i-v_k}(g_k y) = 0, \quad k = 1, \ldots, m, \end{cases}$$

with respective dual problems

$$\mathbb{Q}_i^* \begin{cases} \min\limits_{X \succeq 0, Z_k} -X(1,1) - \displaystyle\sum_{k=1}^m g_k(0) Z_k(1,1) \\ \qquad\qquad \langle X, B_\alpha \rangle + \displaystyle\sum_{k=1}^m \langle Z_k, C_\alpha^k \rangle = (g_0)_\alpha,\ \forall \alpha \neq 0 \end{cases}$$

where X, Z_k are real-valued symmetric matrices, the "dual variables" associated with the constraints $M_i(y) \succeq 0$ and $M_{i-v_k}(g_k y) \succeq 0$ respectively, and where we have written

$$M_i(y) = \sum_\alpha B_\alpha y_\alpha;\ M_{i-v_k}(g_k y) = \sum_\alpha C_\alpha^k y_\alpha,\ k = 1, \ldots, n,$$

for appropriate real-valued symmetric matrices B_α, C_α^k, $k = 1, \ldots, n$.

In the standard terminology, the constraint $M_i(y) \succeq 0$ is called a linear matrix inequality (LMI) and $\mathbb{Q}_i$ and its dual $\mathbb{Q}_i^*$ are so-called positive semidefinite programs, the LMI relaxations of $\mathbb{P}$. The reader interested in more details on SDP and LMIs is referred to [VB96] and the many references therein.

Remark: In the case of 0-1 programming (and more generally, discrete optimization problems) the relaxations $\mathbb{Q}_i$ simplify. Indeed, instead of explicitly stating the LMIs associated with the integrality constraints $x_i^2 = 1$, $i = 1, ..., n$, it suffices to replace (in all the other LMIs) every occurrence of a variable y_α by y_β with $\beta_i = 1$ if $\alpha_i >= 1$, for all $i = 1, ..., n$. This significantly reduces the number of variables in the resulting relaxation $\mathbb{Q}_i$.

Convergence: We make the following assumption on the set $\mathbb{K}$ defined in (6).

Assumption A: $\mathbb{K}$ is compact and there exist a polynomial $u \in \mathbb{R}[x_1, \ldots, x_n]$ of the form

$$x \mapsto u(x) := u_0(x) + \sum_{i=1}^m u_i(x) g_i(x), \tag{7}$$

for some polynomials $\{u_i\}$, all sums of squares, and such that $\{x \in \mathbb{R}^n \mid u(x) \geq 0\}$ is compact.

Assumption A is satisfied in many cases of interest. For instance, it holds as soon as $\{x \in \mathbb{R}^n \mid g_k(x) \geq 0\}$ is compact for some $k \in \{1, \ldots, m\}$, or when all the g_i are linear and $\mathbb{K}$ is compact (hence a convex polytope), for 0-1 (and more generally discrete) programs. In addition, if one knows that a global minimizer x^* of $\mathbb{P}$ satisfies $\|x^*\| \leq M$ for some $M > 0$, then adding the constraint $M - \|x\| \geq 0$ in the definition (6) of $\mathbb{K}$ will ensure that Assumption A holds.

Under Assumption A it was proved in [Las01] that $\inf \mathbb{Q}_i \uparrow \inf \mathbb{P}$ as $i \to \infty$. Moreover, if $g_0(x) - p^*$ has the representation (7), that is,

$$x \mapsto g_0(x) := q_0(x) + \sum_{i=1}^{m} q_i(x) g_i(x),$$

for some polynomials $\{q_i\}$, all sums of squares, and of degree at most $2i_0$, then for all $j \geq i_0$,

$$\sup \mathbb{Q}_j^* = \max \mathbb{Q}_j^* = \min \mathbb{Q}_j = \inf \mathbb{Q}_j = \min \mathbb{P} = p^*.$$

In addition, any optimal solution of $\mathbb{Q}_j$ identifies the vector of moments of a probability measure with mass concentrated on some global minimizers of $\mathbb{P}$.

3 Illustration

In this section we describe a small numerical examples to illustrate the basic use of GloptiPoly. We consider non-convex quadratic problem [Flo99, Pb. 3.5]:

$$\begin{array}{ll} \min & -2x_1 + x_2 - x_3 \\ \text{s.t.} & x_1(4x_1 - 4x_2 + 4x_3 - 20) + x_2(2x_2 - 2x_3 + 9) + x_3(2x_3 - 13) + 24 \geq 0 \\ & x_1 + x_2 + x_3 \leq 4, \quad 3x_2 + x_3 \leq 6 \\ & 0 \leq x_1 \leq 2, \quad 0 \leq x_2, \quad 0 \leq x_3 \leq 3. \end{array}$$

To define this problem with GloptiPoly we use the following Matlab script:

```
>> P = defipoly({'min -2*x1+x2-x3',...
'x1*(4*x1-4*x2+4*x3-20)+x2*(2*x2-2*x3+9)+x3*(2*x3-13)+24>=0',...
'x1+x2+x3<=4', '3*x2+x3<=6',...
'0<=x1', 'x1<=2', '0<=x2', '0<=x3', 'x3<=3'}, 'x1,x2,x3');
```

In the above script, we use features of the Symbolic Math Toolbox version 2.1, the Matlab gateway to the kernel of Maple V [Map01]. It is also possible to enter problems into GloptiPoly without symbolic computations, see [Glo02] for more information.

To solve the first LMI relaxation of the quadratic problem, we type:

```
>> output = gloptipoly(P)
output =
    status: 0
      crit: -6.0000
       sol: {}
```

Field `status = 0` indicates that it is not possible to detect global optimality with this LMI relaxation, hence `crit = -6.0000` is a lower bound on the global optimum.

Next we try to solve the second, third and fourth LMI relaxations of the quadratic problem with the instructions:

```
>> output = gloptipoly(P,2)          >> output = gloptipoly(P,3)
output =                             output =
    status: 0                            status: 0
      crit: -5.6923                        crit: -4.0685
       sol: {}                              sol: {}
>> output = gloptipoly(P,4)
output =
    status: 1
      crit: -4.0000
       sol: {[3x1 double]  [3x1 double]}
>> output.sol{:}
ans =                                ans =
    2.0000                               0.5000
    0.0000                               0.0000
    0.0000                               3.0000
```

Both the second and third LMI relaxations return tighter lower bounds on the global optimum. Eventually global optimality is reached at the fourth LMI relaxation (certified by `status = 1`). GloptiPoly also returns two globally optimal solutions $x_1 = 2$, $x_2 = 0$, $x_3 = 0$ and $x_1 = 0.5$, $x_2 = 0$, $x_3 = 3$ leading to `crit = -4.0000`.

As shown below, the number of LMI variables and the size of the relaxed LMI problem, hence the overall computational time, increase quickly with the relaxation order:

LMI order	1	2	3	4	5	6
LMI optimum	-6.0000	-5.6923	-4.0685	-4.0000	-4.0000	-4.0000
LMI variables	9	34	83	164	285	454
LMI size	24	228	1200	4425	12936	32144

4 Features

As shown by the above numerical example, GloptiPoly is designed to solve an LMI relaxation of a given order, so it can be invoked iteratively with increasing orders until the global optimum is reached. Asymptotic convergence of the optimal values of the LMI relaxations to the global optimal value of the original problem is ensured when the compact set $\mathbb{K}$ of feasible solutions satisfies Assumption A. This condition is satisfied in many practical optimization problems, see [Las01,Las02].

General features of GloptiPoly are listed below:

- Certificate of global optimality
- Automatic extraction of globally optimal solutions
- 0-1 or ± 1 integer constraints on some of the decision variables (combinatorial optimization problems)
- Generation of input and output data in SeDuMi's format
- Generation of moment matrices associated with LMI relaxations
- User-defined scaling of decision variables
- Exploits sparsity of polynomial data.

Finally note that for technical reasons there is currently a limitation on the number of variables handled by GloptiPoly. For example, the current version of GloptiPoly is not able to handle quadratic problems with more than 19 variables. This limitation should be removed soon. For more details, see [Glo02].

5 Performance

All the computations in this section were carried out with Matlab 6.1 and SeDuMi 1.05 with relative accuracy `pars.eps = 1e-9` on a PC with a Pentium IV 1.6 Mhz processor with 512 Mb RAM.

5.1 Continuous optimization problems

We report in Table 1 the performance of GloptiPoly on a series of benchmark non-convex continuous optimization problems. For each problem we indicated the number of decision variables 'var', the number of inequality or equality constraints 'cstr', and the maximum degree arising in the polynomial expressions 'deg'. In almost all reported instances the global optimum was reached exactly by an LMI relaxation of small order, reported in the column entitled 'order'. CPU times are in seconds. 'LMI var' is the dimension of SeDuMi dual vector y, whereas 'LMI size' is the dimension of SeDuMi primal vector x, see [SDM99]. As indicated by the label 'dim' in the rightmost column, quadratic problems 2.8, 2.9 and 2.11 in [Flo99] involve more than 19 variables and could not be handled by the current version of GloptiPoly. Except for problems 2.4 and 3.2, the computational load is moderate.

5.2 Discrete optimization problems

We also report the performance of GloptiPoly on a series of small-size combinatorial optimization problems. In Table 2 we first let GloptiPoly converge to the global optimum, in general extracting several solutions. The number of extracted solutions is reported in the column entitled 'sol'.

Then, we slightly perturbed the criterion to be optimized in order to destroy the problem symmetry. Proceeding this way, the optimum solution is generically unique and convergence to the global optimum is ensured more easily, cf. Table 3. See also [Glo02] for more details on this technique.

5.3 Polynomial systems of equations

Multivariate polynomial systems of equations can be solved with GloptiPoly. We tested its performance on a series of benchmark examples taken from [Ver99] and [Fri00], where we removed examples featuring complex coefficients (recall that GloptiPoly handles real-valued polynomials only). Short descriptions of the benchmarks are given in Tables 4 and 5.

We carried out our experiments by solving feasibility problems, i.e. no criterion was optimized. We did not attempt to count or enumerate all the solutions to the polynomial systems of equations, since this is outside the scope of GloptiPoly. Note that in the absence of a criterion to optimize, GloptiPoly solves the LMI relaxations by minimizing the trace of the moment matrix. Alternative criteria (such as e.g. minimum coordinate or minimum Euclidean-norm solution) are of course possible, but not investigated here.

Our results are reported in Tables 6 and 7. Column 'sol' indicates the number of solutions successfully extracted by GloptiPoly. In the last column the label 'mem' means that the error message 'out of memory' was issued by SeDuMi. GloptiPoly successfully solved about 90% of the systems.

6 Conclusion

GloptiPoly is as a general-purpose software with a user-friendly interface to solve in a unified way a wide range of small- to medium-size non-convex polynomial optimization problems. As illustrated by extensive numerical examples, the main strength of GloptiPoly is that no expert tuning is necessary to cope with very distinct problems coming from different branches of engineering and applied mathematics. GloptiPoly can be used as a black-box software, so it cannot be considered as a competitor to highly specialized codes for solving e.g. sparse polynomial systems of equations or large combinatorial optimization problems.

It is well-known that problems involving polynomial bases with monomials of increasing powers are naturally badly conditioned. If lower and upper bounds on the optimization variables are available as problem data, it may be a good idea to scale all the intervals around one. Alternative bases such as Chebyshev polynomials may also prove useful.

Finally, it would be instructive to compare GloptiPoly with the recently developed software SOSTOOLS [SOS02], also invoking SeDuMi to solve sums of squares optimization programs over polynomials, based on the theory described in [Par00].

Acknowledgment

Work of the first author was partially supported by the Grant Agency of the Czech Republic under Project No. 102/02/0709.

References

[Anj01] M. Anjos. New Convex Relaxations for the Maximum Cut and VLSI Layout Problems. *PhD Thesis*, Waterloo University, Ontario, Canada, 2001. See `orion.math.uwaterloo.ca/~hwolkowi`

[Flo99] C. A. Floudas, P. M. Pardalos, C. S. Adjiman, W. R. Esposito, Z. H. Gümüs, S. T. Harding, J. L. Klepeis, C. A. Meyer, C. A. Schweiger. Handbook of Test Problems in Local and Global Optimization. *Kluwer Academic Publishers*, Dordrecht, 1999. See `titan.princeton.edu/TestProblems`

[Fri00] The Numerical Algorithms Group Ltd. FRISCO - A Framefork for Integrated Symbolic/Numeric Computation. European Commission Project No. 21-024, Esprit Reactive LTR Scheme, 2000. See `www.nag.co.uk/projects/frisco.html`

[GW95] M. X. Goemans, D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, Vol. 42, pp. 1115–1145, 1995.

[Glo02] D. Henrion, J. B. Lasserre. GloptiPoly: Global Optimization over Polynomials with Matlab and SeDuMi. LAAS-CNRS Report No. 02057, Toulouse, France, 2002. Version 2.1 available at `www.laas.fr/~henrion/software/gloptipoly`

[Las01] J. B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, Vol. 11, No. 3, pp. 796–817, 2001.

[Las02] J. B. Lasserre. An Explicit Equivalent Positive Semidefinite Program for 0-1 Nonlinear Programs. *SIAM Journal on Optimization*, Vol. 12, No. 3, pp. 756–769, 2002.

[Map01] Waterloo Maple Software Inc. Maple V release 5. 2001. See `www.maplesoft.com`

[Mat01] The MathWorks Inc. Matlab version 6.1. 2001. See `www.mathworks.com`

[Nes00] Y. Nesterov. Squared functional systems and optimization problems. Chapter 17, pp. 405–440 in H. Frenk, K. Roos, T. Terlaky (Editors). High performance optimization. *Kluwer Academic Publishers*, Dordrecht, 2000.

[Par00] P. A. Parrilo. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. *PhD Thesis*, California Institute of Technology, Pasadena, California, 2000. See `www.control.ethz.ch/~parrilo`

[Put93] M. Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, Vol. 42, pp. 969–984, 1993.

[SA90] H. D. Sherali, W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, Vol. 3, pp. 411–430, 1990.

[SA99] H. D. Sherali, W. P. Adams. A reformulation-linearization technique for solving discrete and continuous nonconvex problems, *Kluwer Academic Publishers*, Dordrecht, 1999.

[Sho87] N. Z. Shor. Quadratic optimization problems. *Tekhnicheskaya Kibernetika*, Vol. 1, pp. 128–139, 1987.

[Sho98] N. Z. Shor. Nondifferentiable Optimization and Polynomial Problems. *Kluwer Academic Publishers*, Dordrecht, 1998.

[SDM99] J. F. Sturm. Using SeDuMi 1.02, a Matlab Toolbox for Optimization over Symmetric Cones. *Optimization Methods and Software*, Vol. 11-12, pp. 625–653, 1999. Version 1.05 available at `fewcal.kub.nl/sturm/software/sedumi.html`

[SOS02] S. Prajna, A. Papachristodoulou, P. A. Parrilo. SOSTOOLS: Sum of Squares Optimization Toolbox for Matlab. California Institute of Technology, Pasadena, USA, 2002. Version 1.00 available at `www.cds.caltech.edu/sostools`

[VB96] L. Vandenberghe, S. Boyd. Semidefinite programming. *SIAM Review*, Vol. 38, pp. 49–95, 1996.

[Ver99] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, Vol. 25, No. 2, pp. 251–276, 1999. Database of polynomial systems available at `www.math.uic.edu/~jan/demo.html`

problem	var	cstr	deg	LMI var	LMI size	CPU	order
[Las01, Ex. 1]	2	0	4	14	36	0.13	2
[Las01, Ex. 2]	2	0	4	14	36	0.13	2
[Las01, Ex. 3]	2	0	6	152	2025	1.13	8
[Las01, Ex. 5]	2	3	2	14	63	0.22	2
[Flo99, Pb. 2.2]	5	11	2	461	7987	11.8	3
[Flo99, Pb. 2.3]	6	13	2	209	1421	1.86	2
[Flo99, Pb. 2.4]	13	35	2	2379	17885	1012	2
[Flo99, Pb. 2.5]	6	15	2	209	1519	1.58	2
[Flo99, Pb. 2.6]	10	31	2	1000	8107	67.7	2
[Flo99, Pb. 2.7]	10	25	2	1000	7381	75.3	2
[Flo99, Pb. 2.8]	20	10	2	-	-	-	dim
[Flo99, Pb. 2.9]	24	10	2	-	-	-	dim
[Flo99, Pb. 2.10]	10	11	2	1000	5632	45.3	2
[Flo99, Pb. 2.11]	20	10	2	-	-	-	dim
[Flo99, Pb. 3.2]	8	22	2	3002	71775	3032	3
[Flo99, Pb. 3.3]	5	16	2	125	1017	1.20	2
[Flo99, Pb. 3.4]	6	16	2	209	1568	1.50	2
[Flo99, Pb. 3.5]	3	8	2	164	4425	2.42	4
[Flo99, Pb. 4.2]	1	2	6	6	34	0.17	3
[Flo99, Pb. 4.3]	1	2	50	50	1926	0.94	25
[Flo99, Pb. 4.4]	1	2	5	6	34	0.25	3
[Flo99, Pb. 4.5]	1	2	4	4	17	0.14	2
[Flo99, Pb. 4.6]	2	2	6	27	172	0.41	3
[Flo99, Pb. 4.7]	1	2	6	6	34	0.20	3
[Flo99, Pb. 4.8]	1	2	4	4	17	0.16	2
[Flo99, Pb. 4.9]	2	5	4	14	73	0.31	2
[Flo99, Pb. 4.10]	2	6	4	44	697	0.58	4

Table 1. Continuous optimization problems. CPU times and LMI relaxation orders required to reach global optima.

problem	var	cstr	deg	LMI var	LMI size	CPU	order	sol
QP [Flo99, Pb. 13.2.1.1]	4	4	2	10	29	0.10	1	1
QP [Flo99, Pb. 13.2.1.2]	10	0	2	385	3136	3.61	2	1
Max-Cut P_1 [Flo99, Pb. 11.3]	10	0	2	847	30976	38.1	3	10
Max-Cut P_2 [Flo99, Pb. 11.3]	10	0	2	847	30976	43.7	3	2
Max-Cut P_3 [Flo99, Pb. 11.3]	10	0	2	847	30976	43.0	3	2
Max-Cut P_4 [Flo99, Pb. 11.3]	10	0	2	847	30976	38.8	3	2
Max-Cut P_5 [Flo99, Pb. 11.3]	10	0	2	-	-	-	4	dim
Max-Cut P_6 [Flo99, Pb. 11.3]	10	0	2	847	30976	43.0	3	2
Max-Cut P_7 [Flo99, Pb. 11.3]	10	0	2	847	30976	44.3	3	4
Max-Cut P_8 [Flo99, Pb. 11.3]	10	0	2	847	30976	43.4	3	2
Max-Cut P_9 [Flo99, Pb. 11.3]	10	0	2	847	30976	49.3	3	6
Max-Cut cycle C_5 [Anj01]	5	0	2	31	676	0.19	3	10
Max-Cut complete K_5 [Anj01]	5	0	2	31	961	0.19	4	20
Max-Cut 5-node [Anj01]	5	0	2	31	676	0.24	3	6
Max-Cut antiweb AW_9^2 [Anj01]	9	0	2	-	-	-	4	dim
Max-Cut 10-node Petersen [Anj01]	10	0	2	847	30976	39.6	3	10
Max-Cut 12-node [Anj01]	12	0	2	-	-	-	3	dim

Table 2. Discrete optimization problems. CPU times and LMI relaxation orders required to reach global optima and extract several solutions.

problem	var	cstr	deg	LMI var	LMI size	CPU	order
QP [Flo99, Pb. 13.2.1.1]	4	4	2	10	29	0.06	1
QP [Flo99, Pb. 13.2.1.2]	10	0	2	847	30976	40.0	3
Max-Cut P_1 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.10	2
Max-Cut P_2 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.03	2
Max-Cut P_3 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.98	2
Max-Cut P_4 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.70	2
Max-Cut P_5 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.41	2
Max-Cut P_6 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.66	2
Max-Cut P_7 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.70	2
Max-Cut P_8 [Flo99, Pb. 11.3]	10	0	2	385	3136	3.33	2
Max-Cut P_9 [Flo99, Pb. 11.3]	10	0	2	385	3136	4.03	2
Max-Cut cycle C_5 [Anj01]	5	0	2	30	256	0.22	2
Max-Cut complete K_5 [Anj01]	5	0	2	31	676	0.28	3
Max-Cut 5-node [Anj01]	5	0	2	30	256	0.22	2
Max-Cut antiweb AW_9^2 [Anj01]	9	0	2	465	16900	12.5	3
Max-Cut 10-node Petersen [Anj01]	10	0	2	385	3136	3.14	2
Max-Cut 12-node [Anj01]	12	0	2	793	6241	29.2	2

Table 3. Discrete optimization problems. CPU times and LMI relaxation orders required to reach global optima with perturbed criterion.

problem	short description
boon	neurophysiology problem
bifur	non-linear system bifurcation
brown	Brown's 5-dimensional almost linear system
butcher	Butcher's system from PoSSo test suite
camera1s	displacement of camera between two positions
caprasse	Caprasse's system from PoSSo test suite
cassou	Cassou-Nogues's system from PoSSo test suite
chemequ	chemical equilibrium of hydrocarbon combustion
cohn2	Cohn's modular equations for special algebraic number fields
cohn3	Cohn's modular equations for special algebraic number fields
comb3000	combustion chemistry example for a temperature of 3000 degrees
conform1	Emiris' conformal analysis of cyclic molecules ($b_{11} = -9$)
conform2	Emiris' conformal analysis of cyclic molecules ($b_{11} = -\sqrt{3}/2$)
conform3	Emiris' conformal analysis of cyclic molecules ($b_{11} = -310$)
conform4	Emiris' conformal analysis of cyclic molecules ($b_{11} = -13$)
cpdm5	5-dimensional system of Caprasse and Demaret
d1	sparse system by Hong and Stahl
des18_3	dessin d'enfant
des22_24	dessin d'enfant
discret3	from PoSSo test suite
eco5	5-dimensional economics problem
eco6	6-dimensional economics problem
eco7	7-dimensional economics problem
eco8	8-dimensional economics problem
fourbar	four-bar mechanical design problem
geneig	generalized eigenvalue problem
heart	heart dipole problem
i1	interval arithmetic benchmark
ipp	six-revolute-joint problem of mechanics
katsura5	problem of magnetism in physics
kinema	robot kinematics problem
kin1	inverse kinematics of an elbow manipulator
ku10	10-dimensional system of Ku
lorentz	equilibrium points of 4-dimensional Lorentz attractor
manocha	intersection of high-degree polynomial curves
noon3	neural network modeled by adaptive Lotka-Volterra system
noon4	neural network modeled by adaptive Lotka-Volterra system
noon5	neural network modeled by adaptive Lotka-Volterra system
proddeco	system with product-decomposition structure
puma	hand position and orientation of PUMA robot
quadfor2	Gaussian quadrature formula with 2 knots and 2 weights over [-1,+1]
quadgrid	interpolating quadrature formula for function defined on a grid

Table 4. Short descriptions of polynomial systems of equations. Part 1.

problem	short description
rabmo	optimal multi-dimensional quadrature formulas
rbpl	generic positions of parallel robot
redeco5	reduced 5-dimensional economics problem
redeco6	reduced 6-dimensional economics problem
redeco7	reduced 7-dimensional economics problem
redeco8	reduced 8-dimensional economics problem
rediff3	3-dimensional reaction-diffusion problem
reimer5	5-dimensional system of Reimer
rose	general economic equilibrium problem
s9_1	small system from constructive Galois theory
sendra	from PoSSo test suite
solotarev	from PoSSo test suite
stewart1	direct kinematic problem of parallel robot
stewart2	direct kinematic problem of parallel robot
trinks	from PoSSo test suite
virasoro	construction of Virasoro algebras
wood	system derived from optimizing the Wood function
wright	Wright's system

Table 5. Short descriptions of polynomial systems of equations. Part 2.

problem	var	cstr	deg	LMI var	LMI size	CPU	order	sol
boon	6	6	4	3002	52864	1220	4	8
bifur	3	3	9	454	8717	8.20	5	2
brown	5	5	5	461	4061	6.27	3	1
butcher	7	7	4	6434	120156	-	4	mem
camera1s	6	6	2	209	952	1.33	2	2
caprasse	4	4	4	209	1285	0.58	3	2
cassou	4	4	8	4844	280151	-	8	mem
chemequ	5	5	3	461	3661	9.48	3	1
chemequs	5	5	3	124	486	6.73	2	1
cohn2	4	4	6	209	1229	0.48	3	1
cohn3	4	4	6	209	1229	0.55	3	1
comb3000	10	10	3	1000	4951	24.6	2	1
conform1	3	3	4	83	430	0.22	3	2
conform2	3	3	4	83	430	0.19	3	2
conform3	3	3	4	285	3766	3.89	5	4
conform4	3	3	4	454	8946	12.2	6	2
cpdm5	5	5	3	125	446	0.24	2	1
d1	12	12	3	-	-	-	3	dim
des18_3	8	8	3	12869	303945	-	4	mem
des22_24	10	10	2	1000	5016	77.2	1	1
discret3	8	8	2	44	89	0.31	1	1

Table 6. Polynomial systems of equations. CPU times and LMI relaxation orders required to reach global optima. Part 1.

problem	var	cstr	deg	LMI vas	LMI size	CPU	order	sol
eco5	5	5	3	461	3661	5.98	3	1
eco6	6	6	3	923	7980	57.4	3	1
eco7	7	7	3	1715	15921	256	3	1
eco8	8	8	3	3002	29565	1310	3	1
fourbar	4	4	4	69	229	0.16	2	1
geneig	6	6	3	923	7602	33.2	3	1
heart	8	8	4	3002	31545	1532	3	2
i1	10	10	3	1000	4366	44.1	2	1
ipp	8	8	2	494	2385	6.42	2	1
katsura5	6	6	2	209	952	0.74	2	1
kinema	9	9	2	714	3520	26.4	2	1
kin1	12	12	3	-	-	-	3	dim
ku10	10	10	2	1000	5016	72.5	2	1
lorentz	4	4	2	209	1705	0.64	2	2
manocha	2	2	8	90	826	1.27	6	1
noon3	3	3	3	83	430	0.22	3	1
noon4	4	4	3	209	1285	0.65	3	1
noon5	5	5	3	461	3241	4.48	3	1
proddeco	4	4	4	69	229	0.11	2	1
puma	8	8	2	3002	35505	1136	3	4
quadfor2	4	4	4	209	1495	0.75	3	2
quadgrid	5	5	5	461	3641	10.52	3	1
rabmo	9	9	5	5004	51703	-	3	mem
rbpl	6	6	3	923	7602	36.9	3	1
redeco5	5	5	2	20	41	0.16	1	1
redeco6	6	6	2	27	55	0.13	1	1
redeco7	7	7	2	35	71	0.14	1	1
redeco8	8	8	2	44	89	0.13	1	1
rediff3	3	3	2	9	19	0.09	1	1
reimer5	5	5	6	6187	264516	-	6	mem
rose	3	3	9	679	16681	79.5	7	2
s9_1	8	8	2	494	2385	5.45	2	1
sendra	2	2	7	65	453	0.34	5	1
solotarev	4	4	3	69	257	0.24	2	1
stewart1	9	9	2	714	3520	20.4	2	2
stewart2	12	10	2	1819	9191	372	2	1
trinks	6	6	3	209	925	0.78	2	1
virasoro	8	8	2	44	89	0.16	1	1
wood	4	3	2	69	527	0.20	2	1
wright	5	5	2	20	41	0.17	1	1

Table 7. Polynomial systems of equations. CPU times and LMI relaxation orders required to reach global optima. Part 2.