

**GloptiPoly - Global Optimization over Polynomials
with Matlab and SeDuMi**

Didier HENRION^{1,2}
Jean-Bernard LASSERRE¹

¹LAAS-CNRS Toulouse
²ÚTIA-AVČR Prague

IMA Minneapolis

March 2003

Brief description

GloptiPoly is written as an open-source, general purpose and user-friendly [Matlab](#) software

Optionally, problem definition made easier with Matlab Symbolic Math Toolbox, gateway to [Maple](#) kernel

Gloptipoly solves small to medium **non-convex** global optimization problems with multivariable real-valued **polynomial** objective functions and constraints

Software and documentation available at

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

Methodology

GloptiPoly builds and solves a **hierarchy** of successive **convex linear matrix inequality (LMI) relaxations** of increasing size, whose optima are **guaranteed** to converge asymptotically to the global optimum



Relaxations are built from LMI formulation of **sum-of-squares (SOS)** decomposition of positive multivariable polynomials

LMIs solved with Jos Sturm's SeDuMi

In practice convergence is ensured **fast**, typically at 2nd or 3rd LMI relaxation

LMI relaxation technique

Polynomial optimization problem

$$\begin{aligned} \min \quad & g_0(x) \\ \text{s.t.} \quad & g_k(x) \geq 0, \quad k = 1, \dots, m \end{aligned}$$

When g^* is the global optimum, SOS representation of positive polynomial

$$g_0(x) - g^* = q_0(x) + \sum_{k=1}^m g_k(x) q_k(x) \geq 0$$

where unknowns $q_k(x)$ are SOS polynomials similar to Karush/Kuhn/Tucker multipliers

Using LMI representation of SOS polynomials successive LMI relaxations are obtained by increasing degrees of sought polynomials $q_k(x)$

Theoretical proof of asymptotic convergence...
..but no tight degree upper bounds (not yet)

LMI relaxations: illustration

Non-convex quadratic problem

$$\begin{aligned} \max \quad & 2x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_1 - 6x_2 + 10 \\ \text{s.t.} \quad & -x_1^2 + 2x_1 \geq 0 \\ & -x_1^2 - x_2^2 + 2x_1x_2 + 1 \geq 0 \\ & -x_2^2 + 6x_2 - 8 \geq 0. \end{aligned}$$

LMI relaxation built by replacing each monomial $x_1^i x_2^j$ with a **new decision variable** y_{ij}

For example, quadratic expression

$$-x_1^2 - x_2^2 + 2x_1x_2 + 1 \geq 0$$

replaced with linear expression

$$-y_{20} - y_{02} + 2y_{11} + 1 \geq 0$$

New decision variables y_{ij} satisfy **non-convex** relations such as $y_{10}y_{01} = y_{11}$ or $y_{20} = y_{10}^2$

LMI relaxations: illustration (2)

Relax these non-convex relations by enforcing LMI constraint

$$M_1^1(y) = \left[\begin{array}{c|cc} 1 & y_{10} & y_{01} \\ \hline y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{array} \right] \geq 0$$

Moment or measure matrix of first order relaxing monomials of degree up to 2

We remove the rank constraint on matrix $M_1^1(y)$

First LMI relaxation of original global optimization problem is given by

$$\begin{aligned} \max \quad & 2y_{20} + 2y_{02} - 2y_{11} - 2y_{10} - 6y_{01} + 10 \\ \text{s.t.} \quad & -y_{20} + 2y_{10} \geq 0 \\ & -y_{20} - y_{02} + 2y_{11} + 1 \geq 0 \\ & -y_{02} + 6y_{01} - 8 \geq 0 \\ & M_1^1(y) \geq 0 \end{aligned}$$

LMI relaxations: illustration (3)

To build second LMI relaxation, we must increase size of moment matrix so that it captures expressions of degrees up to 4

Second order moment matrix reads

$$M_2^2(y) = \begin{bmatrix} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{bmatrix}$$

Constraints are also relaxed with additional variables

Second LMI features feasible set included in first LMI feasible set, thus providing a **tighter** relaxation

$$\max \quad 2y_{20} + 2y_{02} - 2y_{11} - 2y_{10} - 6y_{01} + 10$$

$$\text{s.t.} \quad \left[\begin{array}{c|cc} -y_{20} + 2y_{10} & * & * \\ -y_{30} + 2y_{20} & -y_{40} + 2y_{30} & * \\ -y_{21} + 2y_{11} & -y_{31} + 2y_{12} & -y_{22} + 2y_{12} \end{array} \right] \succeq 0$$

$$\left[\begin{array}{c|cc} \begin{pmatrix} -y_{20} - y_{02} \\ +2y_{11} + 1 \end{pmatrix} & * & * \\ \hline \begin{pmatrix} -y_{30} - y_{12} \\ +2y_{21} + y_{10} \end{pmatrix} & \begin{pmatrix} -y_{40} - y_{22} \\ +2y_{31} + y_{20} \end{pmatrix} & * \\ \begin{pmatrix} -y_{21} - y_{03} \\ +2y_{12} + y_{01} \end{pmatrix} & \begin{pmatrix} -y_{31} - y_{13} \\ +2y_{22} + y_{11} \end{pmatrix} & \begin{pmatrix} -y_{22} - y_{04} \\ +2y_{13} + y_{02} \end{pmatrix} \end{array} \right] \succeq 0$$

$$\left[\begin{array}{c|cc} -y_{02} + 6y_{01} - 8 & * & * \\ \hline \begin{pmatrix} -y_{12} + 6y_{11} \\ -8y_{10} \end{pmatrix} & \begin{pmatrix} -y_{22} + 6y_{21} \\ -8y_{20} \end{pmatrix} & * \\ \begin{pmatrix} -y_{03} + 6y_{02} \\ -8y_{01} \end{pmatrix} & \begin{pmatrix} -y_{13} + 6y_{12} \\ -8y_{11} \end{pmatrix} & \begin{pmatrix} -y_{04} + 6y_{03} \\ -8y_{02} \end{pmatrix} \end{array} \right] \succeq 0$$

$$M_2^2(y) \succeq 0$$

Numerical example (1)

Quadratic problem 3.5 in [Floudas/Pardalos 99]

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

To define this problem with GloptiPoly we use the following Matlab/Maple 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');
```

To solve the [first LMI relaxation](#) 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

Numerical example (2)

Next we try to solve the [second](#), [third](#) and [fourth](#) LMI relaxations

```
>> 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 second and third LMI relaxations return **tighter lower bounds** on the global optimum

Eventually **global optimality** is reached at 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

$$\text{crit} = -4.0000$$

Numerical example (3)

Number of LMI variables (M) and size of relaxed LMI problem (N) increase quickly with relaxation order:

Relaxation	LMI opt	M	N
1	-6.0000	9	24
2	-5.6923	34	228
3	-4.0685	83	1200
4	-4.0000	164	4425
5	-4.0000	285	12936
6	-4.0000	454	32144

..yet fourth LMI relaxation was solved in about 2.5 seconds on a PC Pentium IV 1.6 MHz

Complexity

d : overall polynomial degree ($2\delta = d$ or $d + 1$)

m : number of polynomial constraints

n : number of polynomial variables

M : number of LMI decision variables

N : size of LMI

When d is fixed:

- M grows **polynomially** in $O(n^{2\delta})$
- N grows **polynomially** in $O(n^\delta) + O(mn^{\delta-1})$

When n is fixed:

- M grows **polynomially** in $O(\delta^n)$
- N grows **polynomially** in $O(m\delta^n)$

But when both d and n vary, both M and N grow **exponentially**

$$M = \binom{n + 2\delta}{2\delta} - 1$$
$$N = \binom{n + \delta}{\delta} + m \binom{n + \delta - 1}{\delta - 1}$$

Features

General features of GloptiPoly:

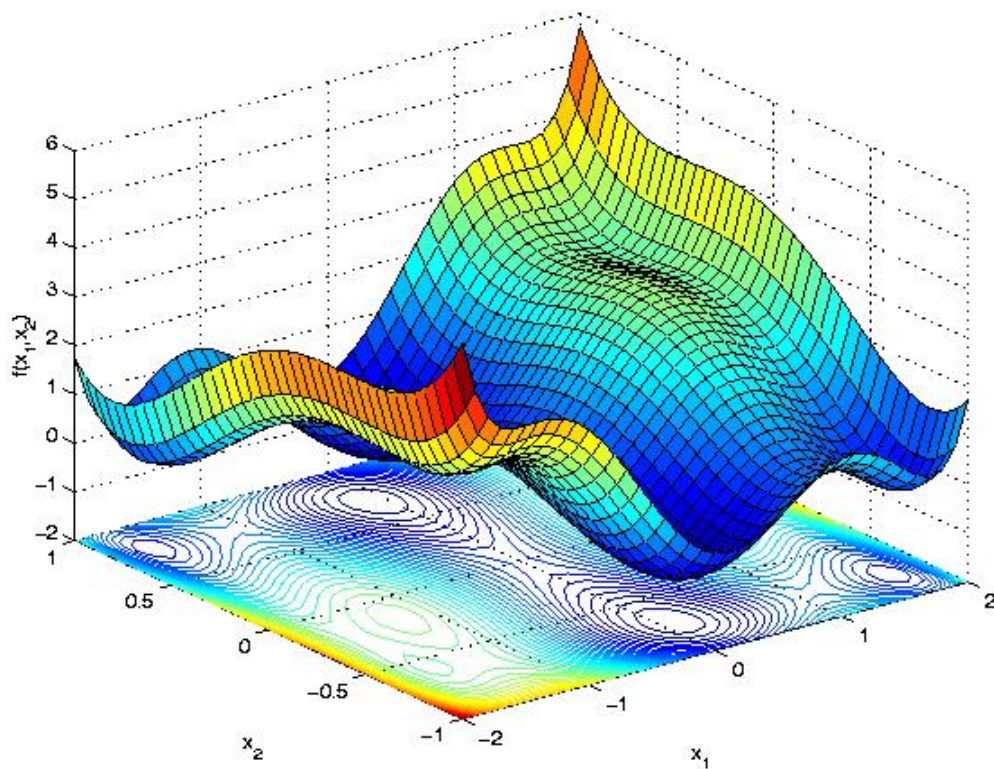
- Certificate of global optimality (rank checks)
- Automatic extraction of globally optimal solutions (multiple eigenvectors)
- 0-1 or ± 1 integer constraints on some of the decision variables (combinatorial optimization problems)
- Generation of input and output data in SeDuMi's format
- Generation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

Benchmark examples

Continuous problems

Mostly from Floudas/Pardalos 1999 handbook

About 80 % of pbs solved with LMI relaxation of small order (typically 2 or 3) in less than 3 seconds on a PC Pentium IV at 1.6 MHz with 512 Mb RAM

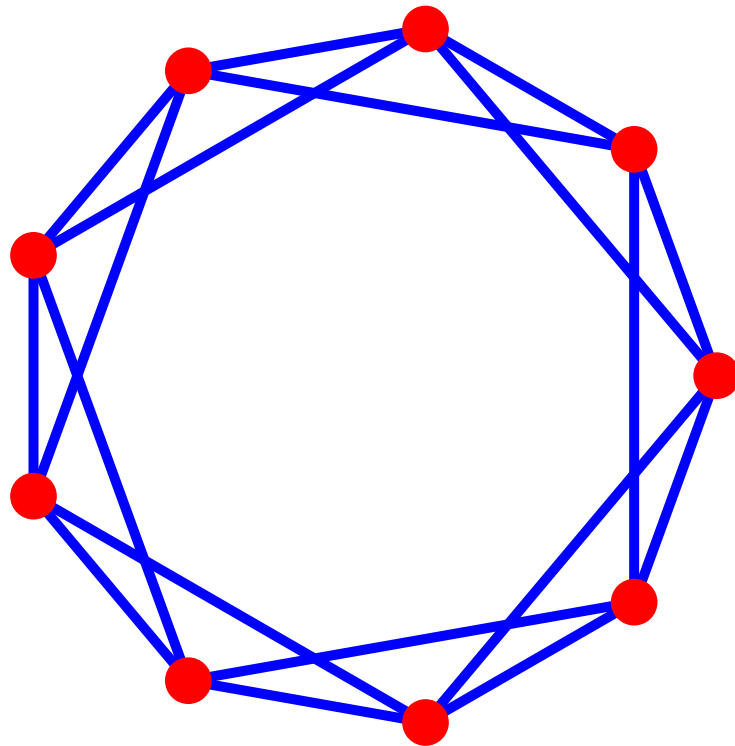


Six-hump camel back function

Benchmark examples Discrete problems

From Floudas/Pardalos handbook and also
Anjos' Ph.D (Univ Waterloo)

By perturbing criterion (destroys symmetry)
global convergence ensured on **80 %** of pbs
in **less than 4 seconds**



MAXCUT on antiweb AW_9^2 graph

Benchmark examples

Polynomial systems of equations

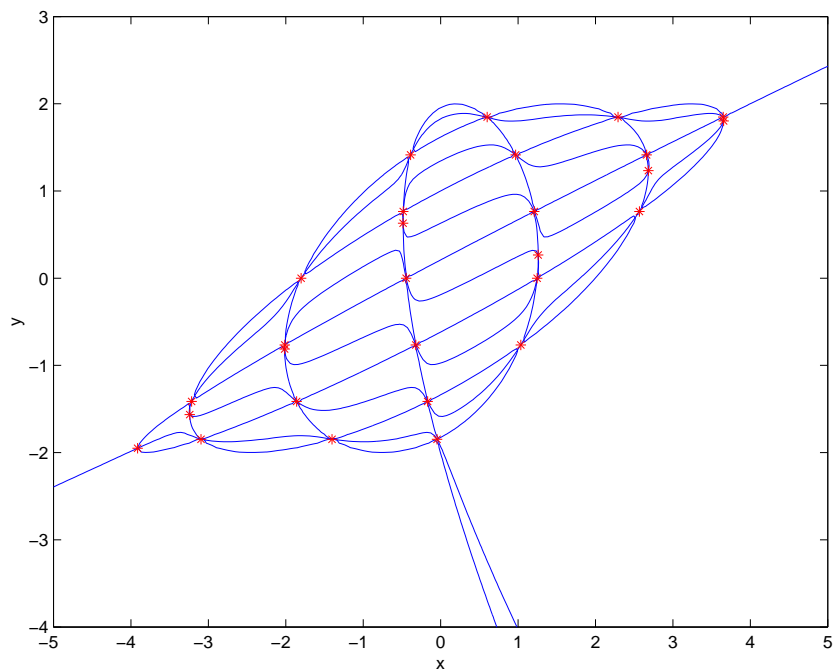
From Verschelde's database and Frisco INRIA project
Real coefficients & solutions only

Out of 59 systems:

- 61 % solved in $t < 10$ secs
- 20 % solved in $10 < t < 100$ secs
- 10 % solved in $t \geq 100$ secs
- 9 % out of memory

No criterion optimized

No enumeration of all solutions



Intersections of seventh and eighth degree polynomial curves

Conclusions

GloptiPoly is a **general-purpose** software with a **user-friendly** interface

Pedagogical flavor, black-box approach, no expert tuning required to cope with **very distinct** applied maths and engineering pbs

Not a competitor to highly specialized codes for solving polynomial systems of equations or large combinatorial optimization pbs

Numerical conditioning (Chebyshev basis) and problem **structure** (Hankel/Toeplitz matrices) deserve further study

See also Parrilo's **SOSTOOLS** software