Solving Global Optimization Problems over Polynomials with GloptiPoly

Didier HENRION[1,2]
Jean-Bernard LASSERRE[1,3]

[1]LAAS-CNRS Univ Toulouse
[2]Czech Tech Univ Prague
[3]IMT Univ Toulouse

9 November 2007

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 multivariate real-valued polynomial objective functions and constraints

Software and documentation available at

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

Metholodogy

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 build from LMI formulation of sum-of-squares (SOS) decomposition of multivariate polynomials

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

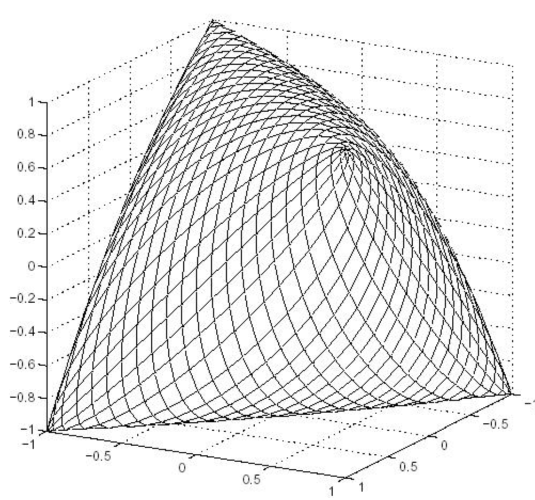
LMI optimization

LMI are solved by convex linear programming problems over the cone of positive semidefinite matrices, also called SDP, or semidefinite programming

Canonical form of an LMI

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \succeq 0$$

where x is a vector of m decision variables and matrices $F_i = F_i^\star$ are given



Linear matrix inequalities

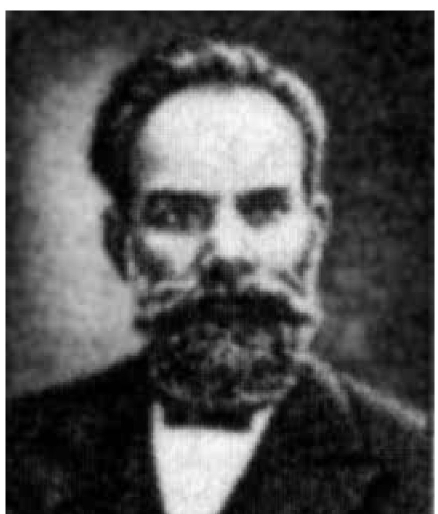
Historically, the first LMIs appeared around 1890 when Lyapunov showed that the differential equation

$$\frac{d}{dt}x(t) = Ax(t)$$

is stable (all trajectories converge to zero) iff there exists a solution to the matrix inequalities

$$A^\star P + PA \prec 0 \quad P = P^\star \succ 0$$

which are linear in unknown matrix P

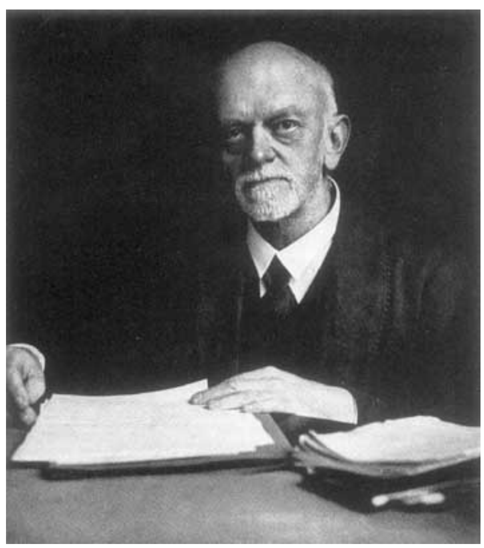


Aleksandr Mikhailovich Lyapunov
(1857 Yaroslavl - 1918 Odessa)

Positive polynomials

The set of univariate polynomials that are positive on the real axis is a convex set that can be described by an LMI

Idea originating from Shor (1987), related with Hilbert's 17th pb about algebraic SOS decompositions



David Hilbert
(1862 Königsberg - 1943 Göttingen)

Can be proved with cone duality (Nesterov) or with theory of moments (Lasserre)

Positive polynomials and LMIs

Example
Global minimization of the polynomial

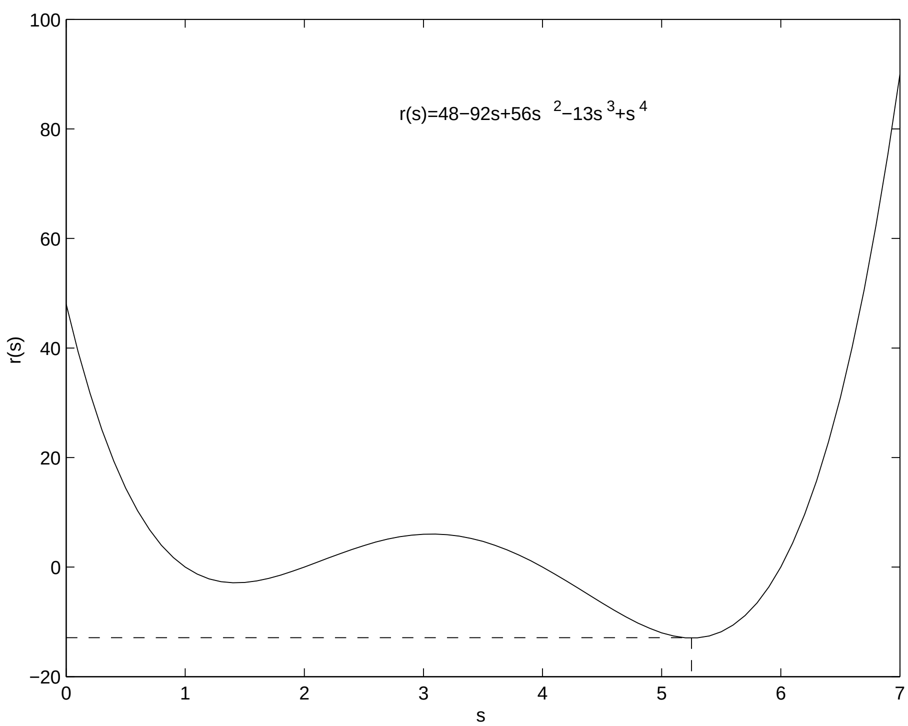
$$r(s) = 48 - 92s + 56s^2 - 13s^3 + s^4$$

Global optimum r^*: maximum value of r_{low} such that $r(s) - r_{\text{low}}$ is a positive polynomial

We just have to solve the LMI

$$\begin{array}{ll} \min & 48 - 92s_1 + 56s_2 - 13s_3 + s_4 \\ \text{s.t.} & \begin{bmatrix} 1 & s_1 & s_2 \\ s_1 & s_2 & s_3 \\ s_2 & s_3 & s_4 \end{bmatrix} \geq 0 \end{array}$$

to obtain $r^* = r(5.25) = -12.89$



Solving LMIs

Development of powerful efficient polynomial-time interior-point algorithms for LP by Karmarkar in 1984

In 1988 Nesterov and Nemirovskii developed interior-point methods that apply directly to LMIs (and even more)

It was then recognized than LMIs can be solved with convex optimization on a home computer

In 1993 Gahinet and Nemirovskii wrote a commercial Matlab package called the LMI Toolbox for Matlab

Several powerful freeware solvers are now available, such as SeDuMi or SDPT3

Applications of LMIs

Various branches of applied mathematics and engineering sciences, including

- Control theory
- Combinatorial optimization
- Signal processing
- Circuit design
- Algebraic geometry
- Machine learning and statistics
- Mechanical structures

Workshops at LAAS-CNRS on LMI/SDP in 2002 and 2004

LMI relaxation technique

Polynomial optimization problem

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

When $g^\star$ is the global optimum, SOS representation of positive polynomial

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

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 convergence

LMI relaxations: illustration

Non-convex quadratic problem

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

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

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

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

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

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

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

Constraints are also relaxed with additional variables

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

$$\begin{array}{ll} \max & 2y_{20} + 2y_{02} - 2y_{11} - 2y_{10} - 6y_{01} + 10 \\ \text{s.t.} & \left[\begin{array}{c|cc} -y_{20} + 2y_{10} & \star & \star \\ \hline -y_{30} + 2y_{20} & -y_{40} + 2y_{30} & \star \\ -y_{21} + 2y_{11} & -y_{31} + 2y_{12} & -y_{22} + 2y_{12} \end{array}\right] \succeq 0 \\ & \left[\begin{array}{c|cc} \left(\begin{array}{c} -y_{20} - y_{02} \\ +2y_{11} + 1 \end{array}\right) & \star & \star \\ \hline \left(\begin{array}{c} -y_{30} - y_{12} \\ +2y_{21} + y_{10} \end{array}\right) & \left(\begin{array}{c} -y_{40} - y_{22} \\ +2y_{31} + y_{20} \end{array}\right) & \star \\ \left(\begin{array}{c} -y_{21} - y_{03} \\ +2y_{12} + y_{01} \end{array}\right) & \left(\begin{array}{c} -y_{31} - y_{13} \\ +2y_{22} + y_{11} \end{array}\right) & \left(\begin{array}{c} -y_{22} - y_{04} \\ +2y_{13} + y_{02} \end{array}\right) \end{array}\right] \succeq 0 \\ & \left[\begin{array}{c|cc} -y_{02} + 6y_{01} - 8 & \star & \star \\ \hline \left(\begin{array}{c} -y_{12} + 6y_{11} \\ -8y_{10} \end{array}\right) & \left(\begin{array}{c} -y_{22} + 6y_{21} \\ -8y_{20} \end{array}\right) & \star \\ \left(\begin{array}{c} -y_{03} + 6y_{02} \\ -8y_{01} \end{array}\right) & \left(\begin{array}{c} -y_{13} + 6y_{12} \\ -8y_{11} \end{array}\right) & \left(\begin{array}{c} -y_{04} + 6y_{03} \\ -8y_{02} \end{array}\right) \end{array}\right] \succeq 0 \\ & M_2^2(y) \succeq 0 \end{array}$$

Numerical example (1)

Quadratic problem 3.5 in [Floudas/Pardalos 99]

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

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

`crit = -4.0000`

Numerical example (3)

Number of LMI variables and size of relaxed LMI problem, hence overall computational time, increase quickly with relaxation order:

Relaxation order	LMI optimum	Number of LMI variables	Size of LMI
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

Features

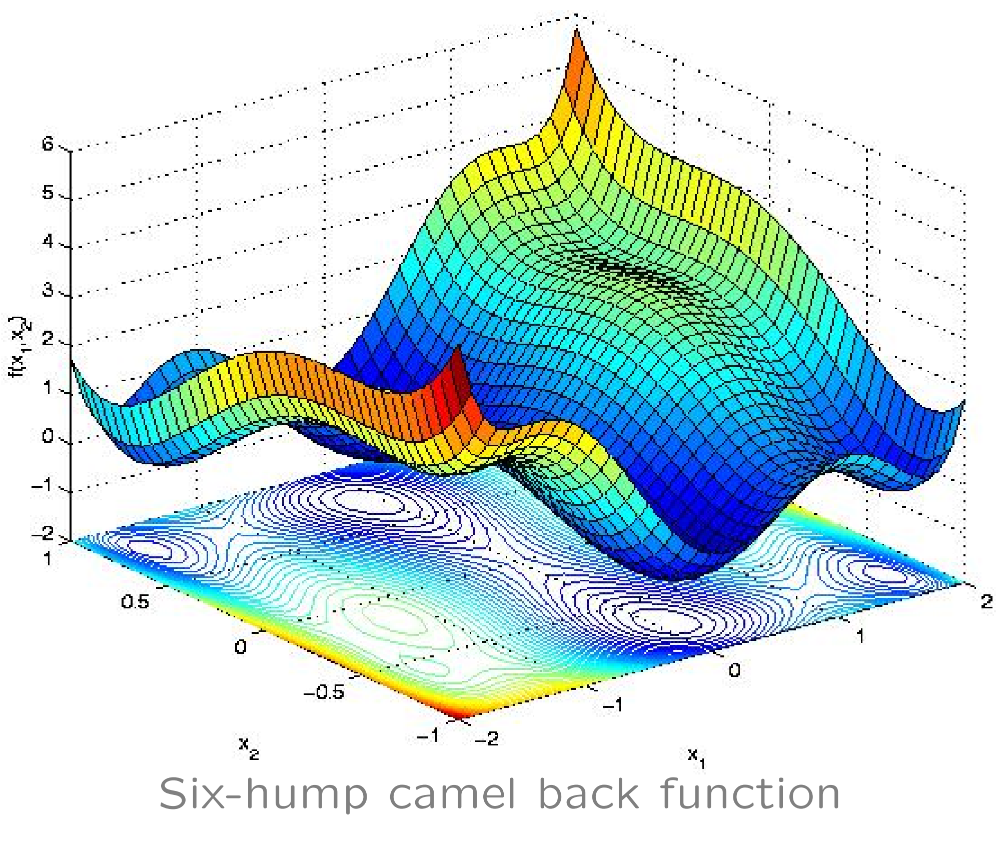
General features of GloptiPoly:

- 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

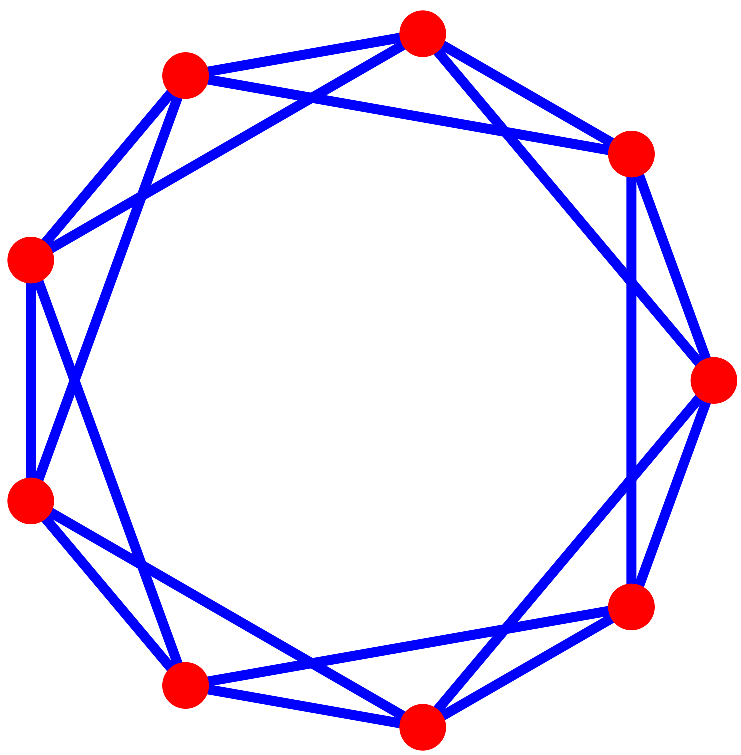


Six-hump camel back function

Benchmark exmaples
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



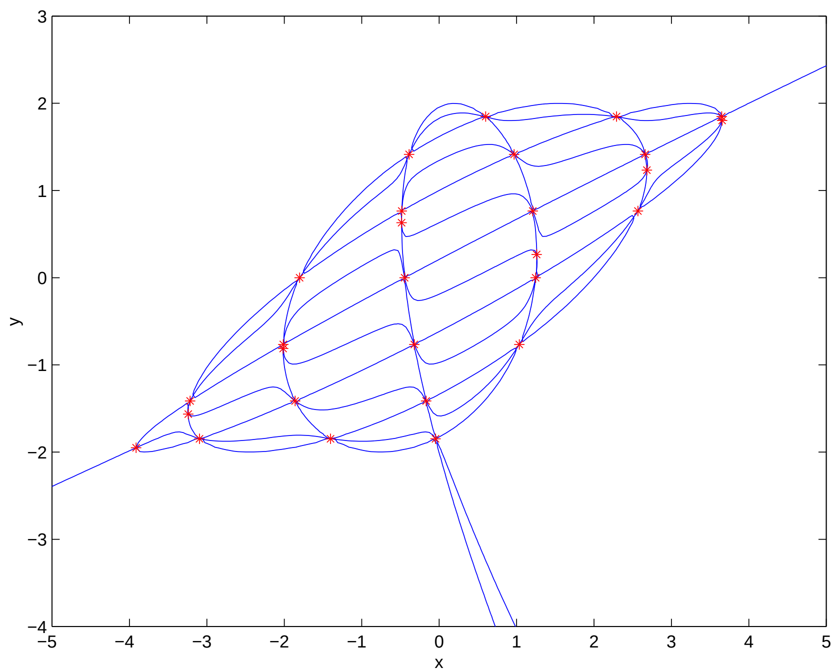
MAXCUT on antiweb AW_9^2 graph

Benchmark examples
Polynomial systems of equations

From Verschelde's and Posso databasis
Real coefficients & coeffs 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) deserves further study