Mathematical Programming: Modelling and Applications

Leo Liberti

LIX, École Polytechnique

liberti@lix.polytechnique.fr

January 2018

Outline

YALMIP: getting started

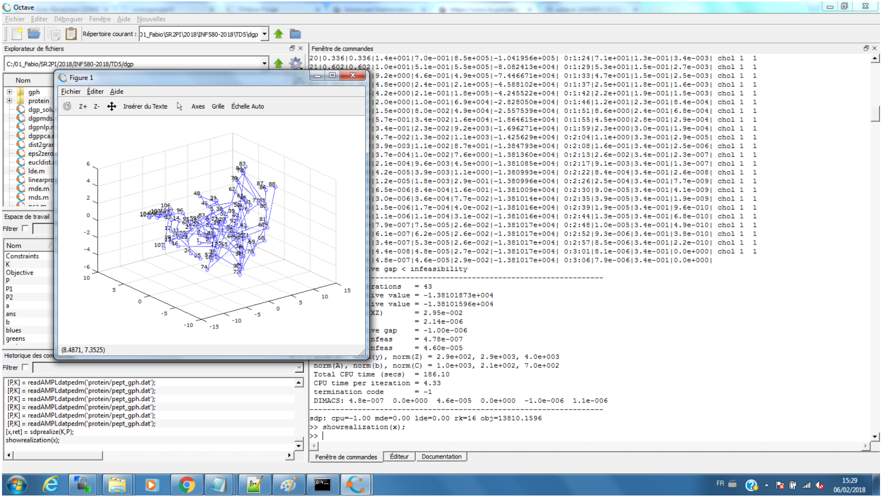
- The YALMIP package is developed with the aim to have a parser that is completely integrated and developed in the MATLAB/OCTAVE environment.
- advantage:platform independence
- drawback: performance loss
- The package is therefore intended for small problems with 10-100 variables and constraints.

YALMIP: installation

Follow these instructions :

`www.lix.polytechnique.fr/~liberti/teaching/dix/inf580-18/README.octave`

YALMIP: installation, test



YALMIP: getting started

- it defines variables using sdpvar
- it defines constraints using Constraint
- it defines objective function using Objective
- it set options via sdpsettings
- it solves the problem using optimize
- it displays solutions Solution or Display

YALMIP: getting started

- A few commands to work with matrices
- Define a *n*x*m* matrix: A = sdpvar(n,m);
- Display its object type: display(A);
- Set its values (e.g. 3x3 matrix): A = [1 2 3; 4 5 6; 7 8 9];
- Display its values: display(value(A));
- Get its transpose: A = A';
- Calculate the 1-norm: norm(A,1);

YALMIP: getting started, a very simple example

```
% Define variables
x = sdpvar(1,1);

% Define constraints
Constraints = [ x<= 10, x>=1];

% Define an objective
Objective = x;

options = sdpsettings('verbose',1,'solver','sdpt3');

% Solve the problem
sol = optimize(Constraints,Objective,options);

% Analyze error flags
if sol.problem == 0
 % Extract and display value
 solution = value(x)
else
 display('Hmm, something went wrong!');
 sol.info
 yalmiperror(sol.problem)
end
```

YALMIP: preliminary exercices

1) Given x, a vector of size 2, minimize the sum of its components, respecting these constraints:

- the first element of the vector is 1;
- the second element is not less than 3;

Display solutions values and objective value.

2) Given x, a vector of size 10, minimize the sum of the product of the transpose of x and x itself plus the 1-norm of x, respecting these constraints:

- the sum of the components of x <= 10;
- the first element of the vector = 0;
- 0.5 <= the second element of the vector <= 1.5;
- For each i between 1 to 7 : x(i) + x(i+1) <= x(i+2) + x(i+3);

Display solutions values and objective value.

SDP

SEMIDEFINITE PROGRAMMING (SDP) refers to optimization problems where decision variables are a symmetric matrix that is required to be semidefinite positive. The standard form is

$$\left.\begin{array}{rrcl} \min & C \bullet X & & \\ \forall i \le m & A^i \bullet X & = & b^i \\ & X & \succeq & 0, \end{array}\right\} \qquad (1)$$

where:

- C and A^i (for $i \le m$) are $n \times n$ symmetric matrices
- X is an $n \times n$ matrix of decision variables,
- $b^i \in \mathbb{R}$ for all $i \le m$
- for two $n \times n$ matrices $L = (\lambda_{ij})$, $M = (\mu_{ij})$ we have $L \bullet M = \sum\limits_{i,j \le n} \lambda_{ij}\mu_{ij}$
- $X \succeq 0$, i.e. X is positive semidefinite

SDP

To make Eq. (1) clearer, write out the componentwise product $\bullet$ of the matrices: $C = (c_{jh})$, $A^i = (a^i_{jh})$ and $X = (x_{jh})$:

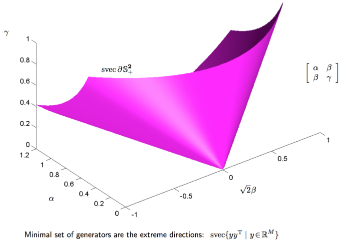
$$\begin{aligned} \min \sum_{j,h \le n} c_{jh} x_{jh} & \\ \forall i \le m \quad \sum_{j,h \le n} a^i_{jh} x_{jh} &= b^i. \end{aligned}$$

This is just an LP subject to a semidefinite constraint $X \succeq 0$. What does $X \succeq 0$ mean?

X, square and symmetric matrix, with real values is PSD if $\forall v \in \mathbb{R}^n$, we have that the scalar $v^T X v \ge 0$.

This is the same as requiring the decision variables x_{jh} to take values such that, when they are arranged in a $n \times n$ array, the resulting matrix A has non-negative eigenvalues.

SDP



This picture of an SDP cone was borrowed from J. Dattorro. Convex Optimization and Euclidean Distance Geometry. $M\epsilon\beta oo$, *PaloAlto*, 2015

SDP : exe. 1

Given the matrices:

$$A_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 7 \\ 1 & 7 & 5 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0 & 2 & 8 \\ 2 & 6 & 0 \\ 8 & 0 & 4 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 9 & 0 \\ 3 & 0 & 7 \end{pmatrix}$$

the parameters $b_1 = 11, b_2 = 19$, **write the corresponding SDP model**.

SDP, exe.2

The standard form of a linear program is:

$$\left.\begin{array}{rrcl} \min & c^T x & & \\ & Ax & = & b \\ & x & \geq & 0, \end{array}\right\} \qquad (2)$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, A an $m \times n$ matrix are the parameters.

Write it as SDP model.

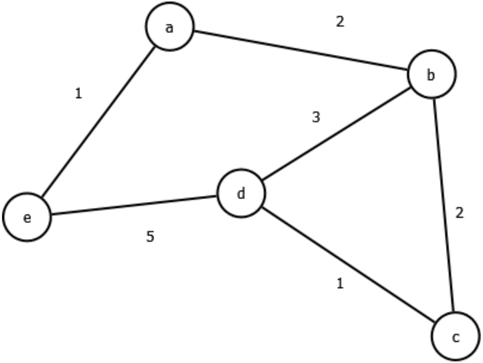
SDP with YALMIP/1

Given a linear dynamic system $x' = Ax$, our goal is to prove stability by finding a symmetric matrix P satisfying:

$$\left.\begin{array}{rcl} A^T P + PA & \preceq & 0 \\ P & \succeq & 0 \end{array}\right\} \qquad (3)$$

- Define a stable matrix A = [-1 2 0 ; -3 -4 1 ; 0 0 -2] and a symmetric matrix P (remember: square matrices are symmetric by default in yalmip).
- $P \succeq 0$ is coded in yalmip as $P \geq 0$
- Impose that the trace of P is equal to 1
- Complete the yalmip script that implements this model
- Display the results

SDP with YALMIP/2



Find a partition of the set of vertices into two parts that maximizes the sum of the weights on the edges that have one end in each part of the partition.
The goal is to remodel the problem using SDP and to implement it using yalmip.
Hint: use a Laplacian matrix as input parameter (and remind that now your decision variable is a matrix).

DGP: SDP relaxation

Reminder: the SDP relaxation of the EDGP system.

$$\left.\begin{array}{rrcl} & \min F \bullet M & & \\ \forall \{u,v\} \in E & X_{uu} - 2X_{uv} + X_{vv} & = & d_{uv}^2 \\ & X & \succeq & 0 \end{array}\right\} \qquad (4)$$

which is a LP subject to a semidefinite constraint $X \succeq 0$

Look at sdprealize.m, which implements DGP as a SDP.

Further Methods

- Inspire yourself with sdprealize.m and create a script ddprealize.m that uses Diagonally Dominant Programming (DDP)

DDP formulation for the DGP

$$\left.\begin{array}{rrcl}
\min & \sum\limits_{\{i,j\}\in E} (X_{ii} + X_{jj} - 2X_{ij}) & & \\
\forall \{i,j\} \in E & X_{ii} + X_{jj} - 2X_{ij} & \geq & d_{ij}^2 \\
\forall i \leq n & \sum\limits_{\substack{j \leq n \\ j \neq i}} T_{ij} & \leq & X_{ii} \\
 & -T \leq X & \leq & T \\
 & T & \geq & 0
\end{array}\right\}$$

- Similarly, create a script ddpdualcone.m using Dual Cone DDP

Barvinok's naive algorithm

```
Given a quadratic feasibility problem:
QF={x in R^Kn : for i in I xQ^ix=b_i} and a solution X
of its SDP relaxation :
RF={X in R^n^2 : for i in I <Q^i,X>=b_i},
retrieve an approximation of x with reasonably
high probability

function x = bvknaivealg(K,X) ....
```