# *Example of convex optimization problem*

Bogdan Dumitrescu

## Optimization problem

- Linear objective, linear and SOC constraints:

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & A_1 x = b_1 \\ & \|A_2 x\| \leq b_2 \end{array} \tag{1}$$

- variable: $x \in \mathbb{R}^n$
- constants: $c \in \mathbb{R}^n$, $A_1 \in \mathbb{R}^{m_1 \times n}$, $b_1 \in \mathbb{R}^{m_2 \times n}$, $A_2 \in \mathbb{R}^{m_2 \times n}$, $b_2 \in \mathbb{R}$

# CVX

- CVX (`http://www.stanford.edu/~boyd/cvx/`) is a Matlab-based modeling system for convex optimization
- Approach: disciplined convex programming
- Natural description of optimization problems
- Uses SeDuMi as solver

## CVX program

```
n = 30;   % number of variables
m1 = 5;  % number of rows of A1
m2 = 30;   % number of rows of A2
c = rand(n,1);
A1 = randn(m1,n);
b1 = randn(m1,1);
A2 = randn(m2,n);
b2 = 1;

cvx_begin  % CVX problem description
  variable x(n)
  minimize( c'*x )
  subject to
    A1*x == b1;
    norm(A2*x) <= b2;
cvx_end
```

## SeDuMi

- SeDuMi (`http://sedumi.mcmaster.ca`) is an SDP solver
- Called from Matlab programs
- Specialized algorithms for LP and SOCP
- The problem must be formulated in one of two standard forms
- Standard equality form:

$$\begin{array}{ll} \min & c_s^T z \\ \text{s.t.} & A_s z = b_s \\ & z \in \mathcal{K} \end{array} \tag{2}$$

where $\mathcal{K}$ is a cone (or a union of cones)

## Transformation to standard form

- Putting $y = A_2 x$ and $t = b_2$, we transform (1) into

$$\begin{aligned} \min \quad & \begin{bmatrix} c^T & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ t \\ y \end{bmatrix} \\ \text{s.t.} \quad & \begin{bmatrix} A_1 & 0 & 0 \\ 0 & 1 & 0 \\ A_2 & 0 & -I \end{bmatrix} \begin{bmatrix} x \\ t \\ y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix} \\ & x \in \mathbb{R}, \ \|y\| \leq t \end{aligned} \tag{3}$$

## SeDuMi program

```
Ks.f = n;       % number of free variables (from x)
Ks.q = m2+1;    % number of variables in SOC constraint
As = [ A1           zeros(m1,1)  zeros(m1,m2)
       zeros(1,n)   1            zeros(1,m2)
       A2           zeros(m2,1) -eye(m2)      ];
bs = [ b1
       b2
       zeros(m2,1) ];
cs = [c; zeros(m2+1,1)];
z = sedumi(As,bs,cs,Ks);
x = z(1:n);
```