Convex Optimization for Information Processing: A State of the Art Review

Wu-Sheng Lu

University of Victoria

Canada

September 3, 2008

Summary

Striking developments in optimization theory have taken place since early 1980's. These include Karmarkar's interior-point methodology for linear programming in the 1980's, its generalization to convex programming in the 1990's, and recent advances in global optimization over polynomials. This body of knowledge has in turn led to widespread application of optimization techniques in information technology fields as well as new solution methods that were considered intractable not too long ago. In this talk we illustrate basic elements of convex programming and survey some current research activities in polynomial optimization. Software tools for convex and polynomial programming will also be discussed.

Outline

- Optimization Problems
- Unconstrained Optimization: An Example
- Unconstrained Optimization: Basic Strategies
- Unconstrained Optimization: Basic Algorithms
- Constrained Optimization: An Example
- Constrained Optimization: KKT Conditions
- Convex Programming: Basic Concepts and Duality
- LP, Convex QP, SDP, and SOCP
- Polynomial Optimization Problems
- Software

1. Optimization Problems

- Unconstrained Optimization

$$\underset{x \in E^n}{\text{minimize}} \quad f(x)$$

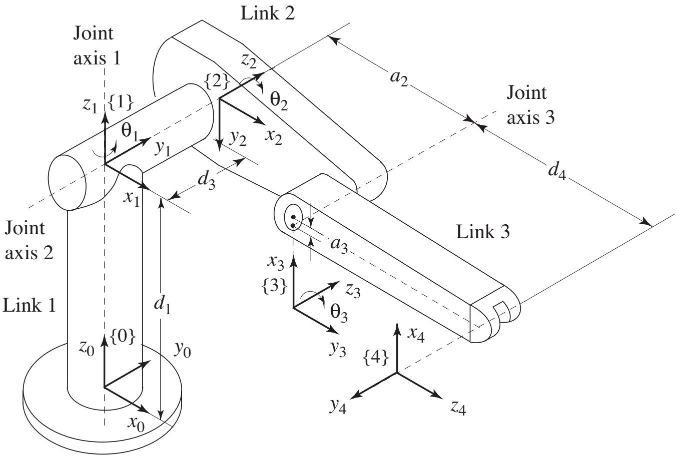
- Constrained Optimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to:} & a_i(x) = 0, \quad i = 1, 2, \ldots, p \\ & c_j(x) \geq 0, \quad j = 1, 2, \ldots, q \end{array}$$

- As long as the objective function and constraints are *quantified*, they do *not* need to be in closed and analytic form.

2. Unconstrained Optimization: An Example

- Inverse kinematics of a robotic manipulator: Given tip position $\{p_x, p_y, p_z\}$, find joint rotations $\{\theta_1, \theta_2, \theta_3\}$.



- A solution via forward kinematics

$$
\begin{aligned}
c_1\left(a_2c_2 + a_3c_{23} - d_4s_{23}\right) - d_3s_1 &= p_x\\
s_1\left(a_2c_2 + a_3c_{23} - d_4s_{23}\right) + d_3c_1 &= p_y\\
d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} &= p_z
\end{aligned}
$$

$$\Updownarrow$$

$$
\begin{aligned}
f_1\left(\Theta\right) &\triangleq c_1\left(a_2c_2 + a_3c_{23} - d_4s_{23}\right) - d_3s_1 - p_x = 0\\
f_2\left(\Theta\right) &\triangleq s_1\left(a_2c_2 + a_3c_{23} - d_4s_{23}\right) + d_3c_1 - p_y = 0\\
f_3\left(\Theta\right) &\triangleq d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} - p_z = 0
\end{aligned}
$$

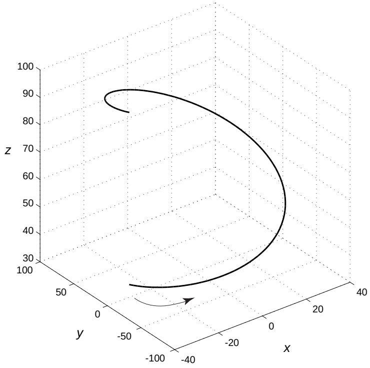
$$\Theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$$

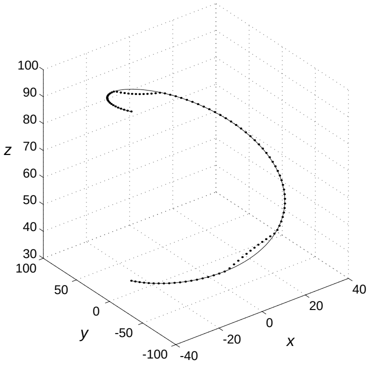
- An optimization-based approach

$$\left\{ \begin{array}{l} f_1(\Theta) = 0 \\ f_2(\Theta) = 0 \\ f_3(\Theta) = 0 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} f_1^2(\Theta) = 0 \\ f_2^2(\Theta) = 0 \\ f_3^2(\Theta) = 0 \end{array} \right\} \Leftrightarrow \sum_{i=1}^{3} f_i^2(\Theta) = 0$$

$$\Rightarrow \qquad \underset{\Theta}{\text{minimize}} \; F(\Theta) = \sum_{i=1}^{3} f_i^2(\Theta)$$

- Advantages of the approach:
  - it works regardless of the relation of the number of equations versus the number of unknowns.
  - it offers a “best” approximate solution if no exact solutions exist.





Desired trajectory (solid line, some parts of it are outside the robot's workspace.)

Actual trajectory (dotted line) produced by the optimization-based solution.

3. Unconstrained Optimization: Basic Strategies

$$\underset{x \in E^n}{\text{minimize}} \quad f(x)$$

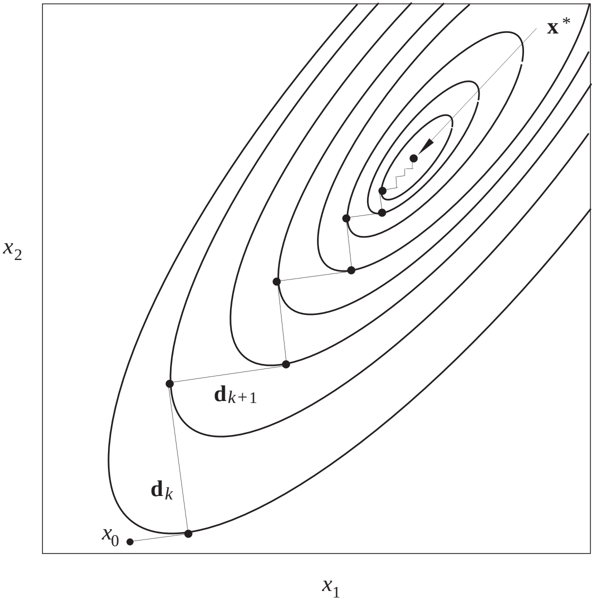
(1) Start with an initial x_0, set $k = 0$ and a tolerance ε.

(2) Compute a search direction d_k

(3) Carry out a line search α_k= arg $\underset{\alpha}{\text{minimize}} \quad f(x_k + \alpha d_k)$

(4) Construct next iterate $x_{k+1} = x_k + \alpha_k d_k$

(5) Check the progress made, $\|\alpha_k d_k\|$, and decide to terminate or to repeat from step (2) with k:= k + 1.


$\mathbf{x}^*$
x_2
$\mathbf{d}_{k+1}$
$\mathbf{d}_k$
x_0
x_1

1. Steepest descent method: $d_k = -\nabla f(x)|_{x=x_k}$
2. Newton's method: $d_k = -\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)\Big|_{x=x_k}$
3. Quasi-Newton methods: $d_k = -S_k \nabla f(x)|_{x=x_k}$

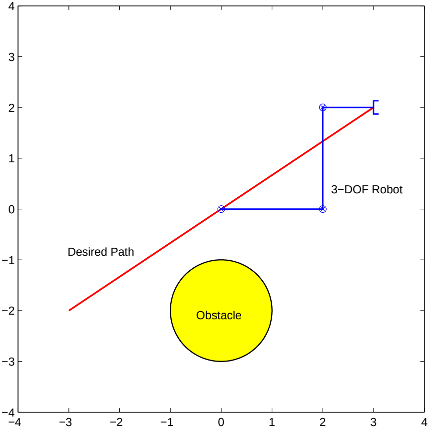
   E.g. the BFGS updating formula:

$$S_0 = I, \quad \delta_k = x_{k+1} - x_k, \quad \gamma_k = g_{k+1} - g_k$$

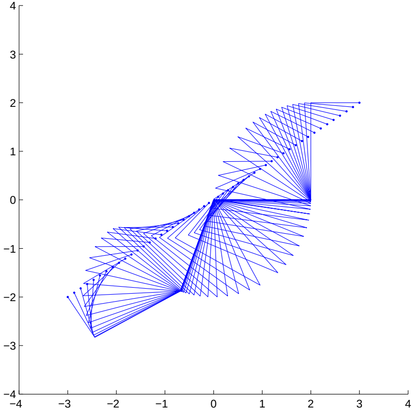
$$S_{k+1} = S_k + \left(1 + \frac{\gamma_k^T S_k \gamma_k}{\gamma_k^T \delta_k}\right) \frac{\delta_k \delta_k^T}{\gamma_k^T \delta_k} - \frac{\delta_k \gamma_k^T S_k + S_k \gamma_k \delta_k^T}{\gamma_k^T \delta_k}$$

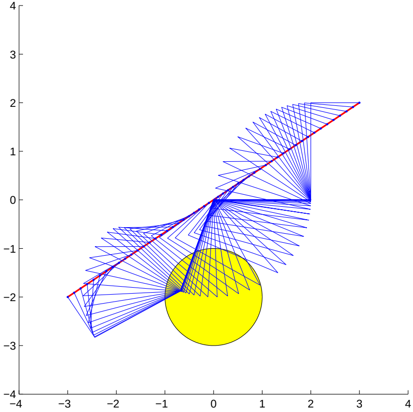
5. Constrained Optimization: An Example

- Global resolution of kinematic redundancy:
  Use the tip of a 3 DOF planar robot to follow a straight path in an environment with an obstacle.



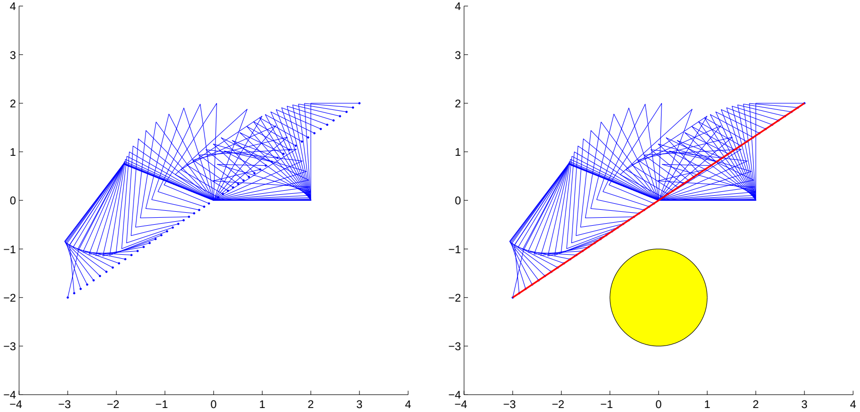
- A typical pseudo-inverse type solution





- A constrained-optimization based path planning for obstacle avoidance

$$\text{minimize} \qquad F = \int_{t_0}^{t_1} g(\theta, t)\, dt$$
$$\text{subject to: } X(t) = f(\theta(t)) \ \ \text{(kinematics)}$$



$$(\mathcal{P}) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to:} & a_i(x) = 0, \quad i = 1, 2, \ldots, p \\ & c_j(x) \geq 0, \quad j = 1, 2, \ldots, q \end{array}$$

- The Karush-Kuhn-Tucker (KKT) necessary conditions for x being a solution of $(\mathcal{P})$:

$$a_i(x) = 0, \quad 1 \leq i \leq p$$

$$c_j(x) \geq 0, \quad 1 \leq j \leq q$$

$$\nabla f(x) = \sum_{i=1}^{p} \lambda_i \nabla a_i(x) + \sum_{j=1}^{q} \mu_j \nabla c_j(x)$$

$$\mu_j \geq 0, \quad \mu_j c_j(x) = 0, \quad 1 \leq j \leq q$$

$$(\mathcal{P}) \qquad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to:} & a_i(x) = 0, \quad i = 1, 2, \ldots, p \\ & c_j(x) \geq 0, \quad j = 1, 2, \ldots, q \end{array}$$

- $(\mathcal{P})$ is called a convex programming (CP) problem if $f(x)$ is a convex function and the feasible region

  $$\text{R} = \left\{ x : a_i(x) = 0 \text{ for } 1 \leq i \leq p;\ \ c_j(x) \geq 0 \text{ for } 1 \leq j \leq p \right\}$$

  is convex.
- For CP problems, the KKT conditions are both necessary and sufficient for x being a solution of $(\mathcal{P})$.

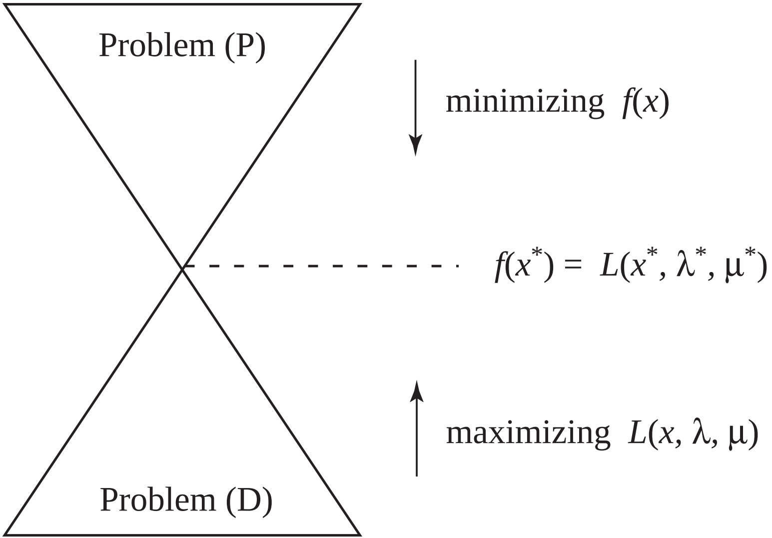
- Wolfe's duality theory for CP

$$(\mathcal{P}) \quad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to:} & a_i(x) = a_i^T x - b_i = 0, \quad i = 1, 2, \ldots, p \\ & c_j(x) \geq 0, \quad\quad\quad\quad j = 1, 2, \ldots, q \end{array}$$

$$\Updownarrow$$

$$(\mathcal{D}) \quad \begin{array}{ll} \underset{x,\lambda,\mu}{\text{maximize}} & L(x, \lambda, \mu) \\ \text{subject to:} & \nabla L(x, \lambda, \mu) = 0 \\ & \mu \geq 0 \end{array}$$

where $L(x, \lambda, \mu) = f(x) - \sum_{i=1}^{p} \lambda_i a_i(x) - \sum_{j=1}^{q} \mu_j c_j(x)$.


Problem (P)
minimizing $f(x)$
$f(x^*) = L(x^*, \lambda^*, \mu^*)$
maximizing $L(x, \lambda, \mu)$
Problem (D)

8. LP, Convex QP, SDP, and SOCP

- There are four classes of CP problems that admit efficient algorithms and software for global solutions:
  - Linear Programming (LP)

$$\text{minimize} \quad c^T x$$
$$\text{subject to: } Ax \geq b$$

  - Convex Quadratic programming (QP)

$$\text{minimize} \quad x^T H x + x^T p$$
$$\text{subject to:} \quad Ax \geq b$$

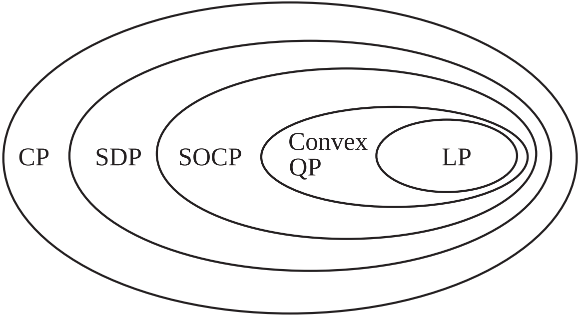
- Semidefinite Programming (SDP)

$$\text{minimize} \quad c^T x$$
$$\text{subject to: } F_0 + x_1 F_1 + \cdots + x_n F_n \succeq 0$$

- Second-Order Cone Programming (SOCP)

$$\text{minimize} \quad c^T x$$
$$\text{subject to: } \left\| A_i^T x + b_i \right\| \leq c_i^T x + d_i, \quad 1 \leq i \leq q$$

- Relation between the four classes of CP problem:



- Solution methods for these CP problems find widespread applications in business, science, and engineering. Below is an example which plays a key role in compressive sensing (CS).

- Find a solution x of an underdetermined linear system $Ax = b$ that has fewest nonzero components. Such a solution is called a sparsest solution.

  - The problem can be described as

$$(\mathcal{L}_0) \quad \begin{array}{ll} \text{minimize} & \|x\|_0 \\ \text{subject to:} & Ax = b \end{array}$$

    where $\|x\|_0 = \sum_{i=1}^{n} |x_i|^0$ = total number of nonzero components in x. Unfortunately, $(\mathcal{L}_0)$ is an NP-hard problem with combinatorial complexity.

- Under mild conditions (Donoho, 2004), $(\mathcal{L}_0)$ is shown to be equivalent to the minimum L_1-norm problem:

$$(\mathcal{L}_1) \quad \begin{array}{l} \text{minimize} \quad \|x\|_1 \\ \\ \text{subject to: } Ax = b \end{array}$$

where $\|x\|_1 = \sum_{i=1}^{n} |x_i|$.

- Note $(\mathcal{L}_1)$ is an LP problem of polynomial complexity.

This is because

$$(\mathcal{L}_1) \quad \begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to:} & Ax = b \end{array}$$

$$\Updownarrow$$

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} \delta_i \\ \text{subject to:} & |x_i| \leq \delta_i, \quad 1 \leq i \leq n \\ & Ax = b \end{array}$$

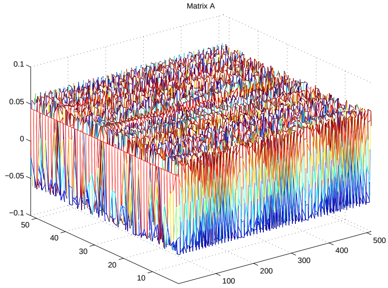
$$\Updownarrow \leftarrow \tilde{x} = \begin{bmatrix} \delta_1 & \cdots & \delta_n & x_1 & \cdots & x_n \end{bmatrix}^T$$

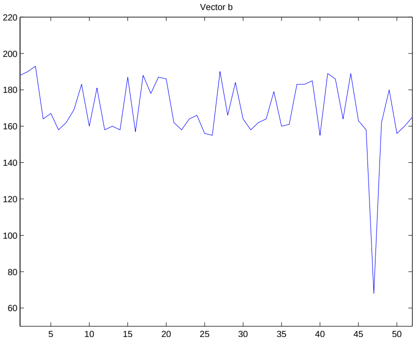
$$\begin{array}{ll} \text{minimize} & c^T \tilde{x} \\ \text{subject to:} & \begin{bmatrix} I_n & -I_n \\ I_n & I_n \end{bmatrix} \tilde{x} \geq 0 \\ & Ax = b \end{array}$$

- Example:

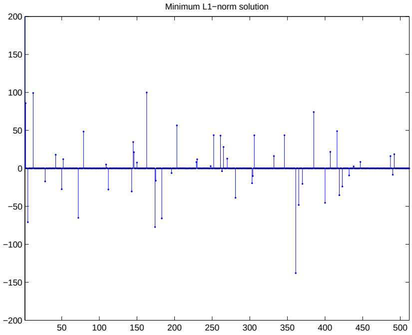
$$(\mathcal{L}_1) \qquad \begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to:} & Ax = b \end{array}$$

where A is constructed by randomly selecting 52 rows from the 512×512 discrete cosine transform (DCT) matrix, and b is formed with 52 pixel values at corresponding positions in a row of text image *boat512*.

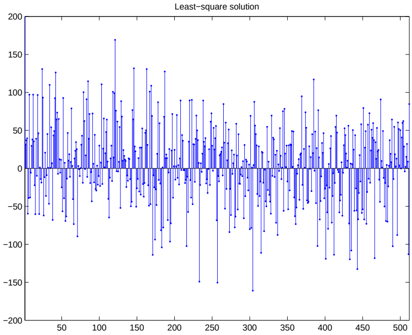




- Minimum L_1-norm solution versus minimum L_2-norm (least-squares) solution:



L_1 solution



L_2 solution

9. Polynomial Optimization Problems (POP)

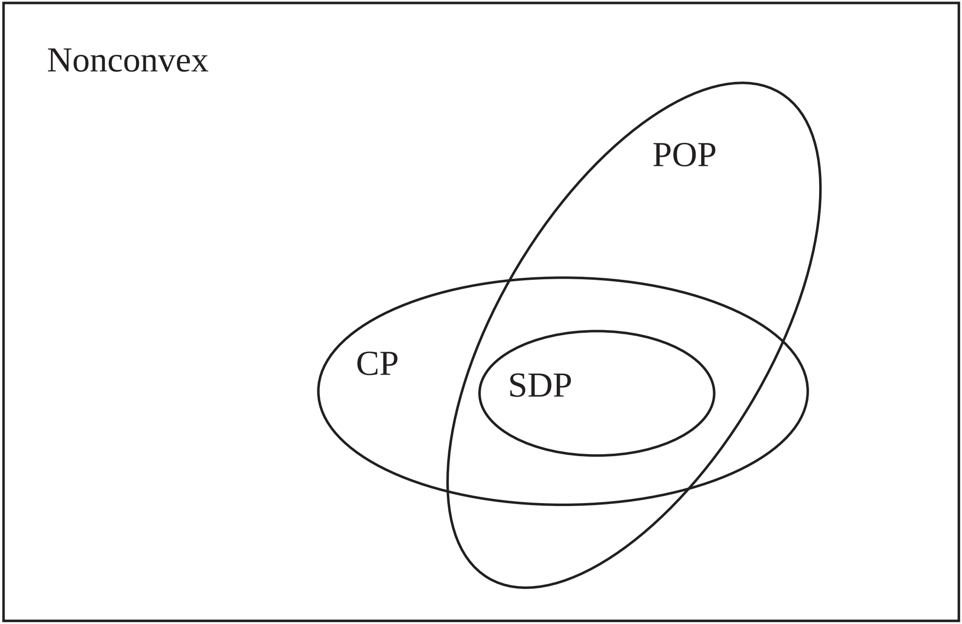
- Unconstrained POP

$$\underset{x \in R^n}{\text{minimize}} \quad p(x)$$

- Constrained POP

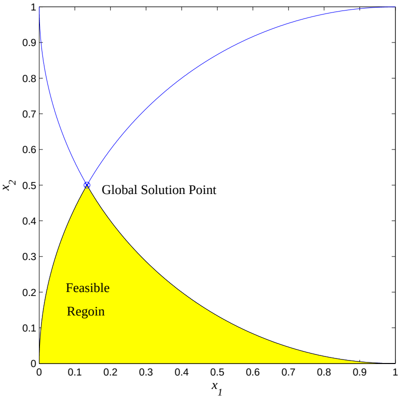
$$\begin{aligned} &\underset{x \in K}{\text{minimize}} \qquad p(x) \\ &K = \{x \in R^n : h_1(x) \geq 0, \ldots, h_m(x) \geq 0\} \end{aligned}$$

- LP $\subseteq$ QP $\subseteq$ SOCP $\subseteq$ SDP $\subseteq$ POP
- POP also includes a great many nonconvex problems!


Nonconvex
POP
CP
SDP

- POP – Example 1: A nonconvex POP

$$\begin{aligned}&\text{minimize} \;\; p(x) = x_1 - 2x_2\\ &\text{subject to:} \;\; x_1 \geq 0, \quad x_2 \geq 0\\ &(x_1 - 1)^2 + x_2^2 \leq 1\\ &(x_1 - 1)^2 + (x_2 - 1)^2 \geq 1\end{aligned}$$



- POP – Example 2: A discrete POP

$$\begin{array}{ll} \text{minimize} & x^T Q x + x^T b \\ \text{subject to:} & x_i \in \{0, 1\}, \quad 1 \le i \le n \end{array}$$

$$\Updownarrow$$

$$\begin{array}{ll} \text{minimize} & x^T Q x + x^T b \\ \text{subject to:} & x_i^2 - x_i = 0, \quad 1 \le i \le n \end{array}$$

$$\Updownarrow$$

$$\begin{array}{ll} \text{minimize} & x^T Q x + x^T b \\ \text{subject to:} & x_i^2 - x_i \ge 0, \quad 1 \le i \le n \\ & -x_i^2 + x_i \ge 0, \quad 1 \le i \le n \end{array}$$

- Solving POP by SDP relaxation: An example

$$\begin{aligned} &\text{minimize} \quad p(x) = x_1 - 2x_2 \\ &\text{subject to: } x_1 \geq 0, \quad x_2 \geq 0, \quad (x_1 - 1)^2 + x_2^2 \leq 1 \\ &\quad (x_1 - 1)^2 + (x_2 - 1)^2 \geq 1 \end{aligned}$$

– a *noncovex* POP with 2 variables and 4 constraints.

- Let $y_{10} = x_1, \quad y_{01} = x_2, \quad y_{20} = x_1^2, \quad y_{02} = x_2^2, \quad y_{11} = x_1 x_2$ and notice the relation between the new variables:

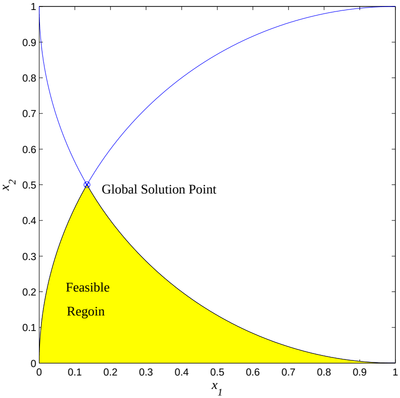
$$\begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_2 \\ x_1 & x_1^2 & x_1 x_2 \\ x_2 & x_1 x_2 & x_2^2 \end{bmatrix} = \begin{bmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{bmatrix} \succeq 0$$

- Now examine the POP problem in extended space:

$$\begin{array}{ll} \text{minimize} & y_{10} - 2y_{01} \\ \text{subject to:} & y_{10} \geq 0, \;\; y_{01} \geq 0, \;\; y_{20} \geq 0, \;\; y_{11} \geq 0, \;\; y_{02} \geq 0 \\ & -y_{20} + 2y_{10} - y_{02} \geq 0 \\ & y_{20} - 2y_{10} + y_{02} - 2y_{01} + 1 \geq 0 \\ & \begin{bmatrix} 1 & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{bmatrix} \succ 0 \end{array}$$

– an SDP problem with 5 variables
$y = \left[\; y_{10} \;\; y_{01} \;\; y_{20} \;\; y_{11} \;\; y_{02} \;\right]^T$ and 8 constraints.

Its solution: $y^* = \begin{bmatrix} 0.1340 & 0.5 & 0.0179 & 0.0670 & 0.25 \end{bmatrix}^T$
which gives
$x^* = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} y_{10} & y_{01} \end{bmatrix}^T = \begin{bmatrix} 0.1340 & 0.5 \end{bmatrix}^T$
– the global solution of the nonconvex POP:



- Theoretical foundation of this SDP relaxation for global POP solutions can be found in
  - J. B. Lasserre, “Global optimization with polynomials and the problem of moments,” SIAM J. Optim., vol. 11, no. 3, pp. 796-817, 2001.
  - J. B. Lasserre, “An explicit equivalent positive semidefinite program for nonlinear 0-1 programs,” SIAM J. Optim., vol. 12, no. 3, pp. 756-769, 2002.

10. Software

- Commercial
  - Optimization Toolbox (MathWorks) – LP, QP
  - Robust Control Toolbox (MathWorks) – SDP
- Public-domain
  - SDPT3 (Cornell, NUS, CMU) – LP, QP, SOCP, SDP
  - SDPA (Tokyo Inst. Tech.) – LP, QP, SOCP, SDP
  - SeDuMi (J.F. Sturm, McMaster) – LP, QP, SOCP, SDP

    http://sedumi.mcmaster.ca
  - GloptiPoly, version 3, (Herion and Lasserre) – POP
  - SparsePOP (Tokyo Inst. Tech.) – POP

- A recent book on optimization :
  A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithm and Engineering Applications*, Springer, New York, 2007.

  MATLAB functions for a variety of unconstrained and constrained optimization algorithms are available in the book's website for download:

  http://www.ece.uvic.ca/˜optimization/

END