Lecture 6

Modified Sum-of-Squares Relaxations for Large Scale Optimizations

Fall 2019

Ashkan Jasour
Moment-SOS Relaxations

Nonlinear-Nonconvex Optimization

Convexification

- SOS Relaxation
- Moment Relaxation

Dual optimization

Semidefinite Program
**Moment-SOS Relaxations: Applications in Robotics and Control**

### Motion Planning
- A. Majumdar, R. Tedrake, “Funnel libraries for real-time robust feedback motion planning”, international journal of robotics and research (IJRR), Volume: 36 issue: 8, page(s): 947-982, 2017

### Planning and Controllers for UAV

### Legged Robots

### Real-Time Planning

### Controller Design
Moment-SOS Relaxations: Applications in Robotics and Control

Validation


- S. Shen, R. Tedrake, “Compositional Verification of Large-Scale Nonlinear Systems via Sums-of-Squares Optimization”, American Control Conference (ACC) 2018

Environment Representation


Control and Analysis

- M. Korda, D. Henrion, C. N. Jones. Controller design and region of attraction estimation for nonlinear dynamical systems. , October 2013, updated in March 2014,


Moment-SOS Relaxations

Nonlinear-Nonconvex Optimization $\rightarrow$ Convexification $\rightarrow$ Semidefinite Program

- SOS Relaxation
- Moment Relaxation

What is the cost of convexification?
Nonlinear Optimization: variables \((x_1, x_2)\)

\[
P^* = \min_{x \in \mathbb{R}^2} \frac{1}{3}x_1^6 - \frac{21}{10}x_1^4 + 4x_1^2 + x_1x_2 + 4x_2^4 - 4x_2^2 + \frac{3}{2}
\]

subject to \(x \in K = \{x \in \mathbb{R}^2 : -\frac{1}{16}x_1^4 + \frac{1}{4}x_1^3 - \frac{1}{4}x_1^2 - \frac{9}{100}x_2^2 + \frac{29}{400} \geq 0\}\)
Nonlinear Optimization: variables \((x_1, x_2)\)

\[
P^* = \min_{x \in \mathbb{R}^2} \frac{1}{3} x_1^6 - \frac{21}{10} x_1^4 + 4x_1^2 + x_1 x_2 + 4x_2^4 - 4x_2^2 + \frac{3}{2}
\]

subject to \(x \in K = \{x \in \mathbb{R}^2 : -\frac{1}{16} x_1^4 + \frac{1}{4} x_1^3 - \frac{1}{4} x_1^2 - \frac{9}{100} x_2^2 + \frac{29}{400} \geq 0\}\)

Moment SDP: variables are moments \(y_{\alpha_1 \alpha_2} = E[x_1^{\alpha_1} x_2^{\alpha_2}]\) \(y = [y_\alpha, \alpha = 0, ..., 6]\)

\[
P^{*\text{mom}} = \min_{y} \frac{1}{3} y_{60} - \frac{21}{10} y_{40} + 4y_{20} + y_{11} + 4y_{01} - 4y_{02} + \frac{3}{2} y_{00}
\]

subject to \(y_{60} = 1\)

\(M_3(y) \succeq 0, M_{3-2}(yy) \succeq 0\)

- Number of Moments in \(\mathbb{R}^n\) up to order 2d:

\[
\binom{n+2d}{n} = \frac{(2d+6)!}{2!6!} = 28
\]
**Nonlinear Optimization:** variables \((x_1, x_2)\)

\[
P^* = \text{minimize}_{x \in \mathbb{R}^2} \quad \frac{1}{3} x_1^6 - \frac{21}{10} x_1^4 + 4 x_1^2 + x_1 x_2 + 4 x_2^4 - 4 x_2^2 + \frac{3}{2}
\]

subject to \(x \in K = \{x \in \mathbb{R}^2 : -\frac{1}{16} x_1^4 + \frac{1}{4} x_1^3 - \frac{1}{4} x_1^2 - \frac{9}{100} x_2^2 + \frac{29}{400} \geq 0\}\)

**Interior-point method**

**Moment SDP:** variables are moments \(y_{\alpha_1 \alpha_2} = E[x_1^{\alpha_1} x_2^{\alpha_2}]\) \(y = [y_{\alpha}, \alpha = 0, ..., 6]\)

\[
P^*_{\text{mom}} = \text{minimize}_{y} \quad \frac{1}{3} y_{60} - \frac{21}{10} y_{40} + 4 y_{20} + y_{11} + 4 y_{01} - 4 y_{02} + \frac{3}{2} y_{00}
\]

subject to \(y_{60} = 1\)

\(M_3(y) \succeq 0, \ M_{3-2}(yy) \succeq 0\)

- Number of Moments in \(\mathbb{R}^n\) up to order \(2d\):

\[
\binom{n+2d}{n} = \frac{(2+6)!}{2!6!} = 28
\]

**SOS SDP:** variables are coefficients of polynomial

\[
P^*_{\text{sos}} = \text{maximize}_{\gamma \in \mathbb{R}, \sigma_1} \quad \gamma
\]

subject to \(p(x) - \gamma - \sigma_1(x) g(x) \in \text{SOS}_6\)

\(\sigma_1(x) \in \text{SOS}_2\)

- Number of coefficients of a \(2d\)-degree polynomial in \(\mathbb{R}^n\):

\[
\binom{n+2d}{n} = \frac{(2+6)!}{2!6!} = 28
\]
What is the cost of convexification?

Convexification increases the dimension of the search space.

- Number of variables of the original nonlinear optimization: $n$
- Number of variables Moment SDP: $\binom{n+2d}{n}$
Moment-SOS Relaxations

What is the cost of convexification?

Convexification increases the **dimension** of the search space.

- **Cost of solving challenging problems**
  - Number of variables of the original nonlinear optimization: \( n \)
  - Number of variables Moment SDP: \( \binom{n+2d}{n} \)

Pros:
- Moment-SOS relaxations solve difficult and challenging mathematical problems.
- They provide insights into challenging problems where no other solid and comprehensive approach exist. (e.g., existing approaches for **nonlinear robust and chance constrained optimizations** work for particular class of problems,...).
Moment-SOS Relaxations

- Current SDP solvers are interior-point based solvers.

- In the absence of problem structure, sum of squares problems are currently limited, roughly speaking, to a several thousands variables (variables in SDP).

- How to address large scale problems?
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.

Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.

   Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)

2) Take advantage of structure of the problem (sparsity) to generate smaller SDP’s.

   Approaches:
   i) Spars Sum-of-Squares Optimization (SSOS)
   ii) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.
   Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)

2) Take advantage of structure of the problem (sparsity) to generate smaller SDP’s.
   Approaches:
   i) Spars Sum-of-Squares Optimization (SSOS)
   ii) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)

3) Efficient Algorithms for Large Scale SDP’s (Lecture 9)
Moment-SOS Relaxations

How to address large scale problems?

1) Modified SOS optimization to generate i) smaller SDP’s or ii) other types of convex constraints like LP.

   Approaches:
   i) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS),
   ii) Bounded degree SOS (BSOS)

2) Take advantage of structure of the problem (sparsity) to generate smaller SDP’s.

   Approaches:
   i) Spars Sum-of-Squares Optimization (SSOS)
   ii) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)

3) Efficient Algorithms for Large Scale SDP’s (Lecture 9)

4) Reformulate original optimization problem to reduce the size of the optimization (Lectures 10 and 11)
Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
Modified SOS optimization that results in LP and Second order cone program

Applications:
Control and analyze of high dimensional systems

Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Sparse Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.


Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Sparse Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3

Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Sparse Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3
(Scaled) Diagonally-Dominant SOS Optimization
(DSOS, SDSOS)
Nonlinear Optimization and Nonnegative polynomials

**Unconstrained Optimization:**

\[
\begin{align*}
\text{minimize} & \quad p(x) \\
\text{subject to} & \quad p(x) \in \mathbb{R}[x]
\end{align*}
\]

**Constrained Optimization:**

\[
\begin{align*}
\text{minimize} & \quad p(x) \\
\text{subject to} & \quad g_i(x) \geq 0, \quad i = 1, \ldots, m \\
p(x), g_i(x) & \in \mathbb{R}[x], \quad i = 1, \ldots, m
\end{align*}
\]
Sum of squares Polynomials

Polynomial \( p(x) \) is **sum of squares** (SOS) polynomial if:

it can be written as a finite sum of squares of other polynomials.

\[
p(x) = \sum_{i=1}^{\ell} h_i^2(x)
\]

\( h_i(x) \in \mathbb{R}[x], \quad i = 1, \ldots, \ell \)

• If polynomial \( p(x) \) is SOS, then it is \( p(x) \geq 0 \) for all

PSD Matrix representation of SOS polynomials

\[
p(x) = B(x)^T Q B(x)
\]

\( Q \in \mathbb{S}^n, \quad Q \succeq 0 \)

*where* \( B(x) \): vector of monomials in \( x \)

*Nonnegative Polynomials*

*SOS Polynomials*
Sum of squares Polynomials

\[ p(x) \in SOS \quad \iff \quad Q \in S^n, \quad Q \succeq 0 \quad \iff \quad \text{Nonnegative Eigenvalues} \quad \iff \quad \text{SDP} \]

PSD Matrix
Sum of squares Polynomials

\[ p(x) \in SOS \quad \text{if and only if} \quad Q \in S^n, \quad Q \succeq 0 \quad \text{has nonnegative eigenvalues} \quad \Rightarrow \quad \text{SDP} \]

- To avoid SDP and obtain \textit{computationally cheap} convex optimizations, we obtain \textit{relaxed condition} for PSD matrices.

- For this, we use the following Results:
  1) Gershgorin Circle Theorem
  2) Diagonally Dominant Matrix (dd)
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[ \text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \]
\[ \text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \]
\[ \text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|) \]
Gershgorin Circle Theorem

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3}$$

$$\text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|)$$
$$\text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|)$$
$$\text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|)$$

- Eigenvalue of $Q$ lies within the Gershgorin discs.

$$Q \in \mathbb{R}^{n \times n} \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n$$
Gershgorin Circle Theorem

Let 

\[
Q = \begin{bmatrix}
Q_{11} & Q_{12} & Q_{13} \\
Q_{21} & Q_{22} & Q_{23} \\
Q_{31} & Q_{32} & Q_{33}
\end{bmatrix} \in \mathbb{R}^{3 \times 3}
\]

Then the Gershgorin discs are defined as:

\[
\text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \\
\text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \\
\text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|)
\]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[
Q \in \mathbb{R}^{n \times n} \quad \Rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n
\]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.
**Gershgorin Circle Theorem**

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[ \text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \]
\[ \text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \]
\[ \text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|) \]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \Rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, ..., n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[
\begin{align*}
\text{Disk}_1(Q_{11}, R_1) &= |Q_{12}| + |Q_{13}| \\
\text{Disk}_2(Q_{22}, R_2) &= |Q_{21}| + |Q_{23}| \\
\text{Disk}_3(Q_{33}, R_3) &= |Q_{31}| + |Q_{32}| 
\end{align*}
\]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \implies \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \in \mathcal{S}^3 \]

\[ (Q_{33} - R_3) \]

Smallest Eigenvalue \[ \geq \min_{i=1,2,3} (Q_{ii} - R_i) \]
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\begin{align*}
\text{Disk}_1(Q_{11}, R_1) &= |Q_{12}| + |Q_{13}| \\
\text{Disk}_2(Q_{22}, R_2) &= |Q_{21}| + |Q_{23}| \\
\text{Disk}_3(Q_{33}, R_3) &= |Q_{31}| + |Q_{32}| \\
\end{align*}

Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \iff \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{i \neq j} |Q_{ij}|), \quad i = 1, \ldots, n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \in \mathbb{S}^3 \]

\[ (Q_{33} - R_3) \quad Q_{33} \quad Q_{11} \quad 0 \quad Q_{22} \]

PSD \rightarrow \text{Smallest Eigenvalue} \geq \min_{i=1,2,3} (Q_{ii} - R_i) \geq 0
Eigenvalue of $Q$ lies within the Gershgorin discs.

We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3}$$

- $Disk_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|)$
- $Disk_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|)$
- $Disk_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|)$

$$Q \in \mathbb{R}^{n \times n} \quad \Leftrightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} Disk_i(Q_{ii}, R_i = \sum_{j \neq i} |Q_{ij}|), \ i = 1, \ldots, n$$

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \in \mathbb{S}^3$$

$$\text{(Q}_{33} - R_3) \quad Q_{33} \quad Q_{11} \quad 0 \quad Q_{22}$$

$$Q_{11} \geq R_1 = |Q_{12}| + |Q_{13}|$$
$$Q_{22} \geq R_2 = |Q_{12}| + |Q_{23}|$$
$$Q_{33} \geq R_3 = |Q_{13}| + |Q_{23}|$$

PD \rightarrow \text{Smallest Eigenvalue} \geq \min_{i=1,2,3} (Q_{ii} - R_i) \geq 0$$
Gershgorin Circle Theorem

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3} \]

\[ \text{Disk}_1(Q_{11}, R_1 = |Q_{12}| + |Q_{13}|) \]
\[ \text{Disk}_2(Q_{22}, R_2 = |Q_{21}| + |Q_{23}|) \]
\[ \text{Disk}_3(Q_{33}, R_3 = |Q_{31}| + |Q_{32}|) \]

- Eigenvalue of \( Q \) lies within the Gershgorin discs.

\[ Q \in \mathbb{R}^{n \times n} \quad \rightarrow \quad \text{Eigenvalues} \in \bigcup_{i=1}^{n} \text{Disk}_i(Q_{ii}, R_i = \sum_{j \neq i} |Q_{ij}|), \ i = 1, ..., n \]

- We use Gershgorin Circle Theorem to obtain relaxed PSD condition in terms of entries of matrices.

\[ Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \in S^3 \]

\[ Q_{33} - R_3 \]

PSD \( \rightarrow \) Smallest Eigenvalue \( \geq \min_{i=1,2,3} (Q_{ii} - R_i) \geq 0 \)

Diagonally Dominant Matrix (dd):

\[ Q \in S^n \quad Q_{ii} \geq \sum_{j \neq i} |Q_{ij}|, \ i = 1, ..., n \quad Q \in S^dd_n \subset S^+_n \]
Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in SOS \quad \text{PSD Matrix} \quad Q \in S^n_+ \quad \text{Nonnegative Eigenvalues} \quad \text{SDP} \]

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in \text{SOS} \]

\[ Q \in \mathcal{S}_+^n \]

\[ \text{Nonnegative Eigenvalues} \]

\[ \text{SDP} \]

\[ Q \in \mathcal{S}_{dd}^n \]

\[ Q_{ii} \geq \sum_{i \neq j} |Q_{ij}|, \quad i = 1, \ldots, n \]

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in \text{SOS} \]

\[ Q \in \mathcal{S}_+^n \]

Nonnegative Eigenvalues

SDP

Relaxation

Relaxation

\[ Q \in \mathcal{S}_d^d \]

Diagonally Dominant Matrix

Linear Constraints

\[ Q_{ii} \geq \sum_{i \neq j} |Q_{ij}|z_{ij}, \quad i = 1, \ldots, n \]

\[ \left\{ \begin{array}{l}
Q_{ii} \geq \sum_{i \neq j} z_{ij}, \quad i = 1, \ldots, n \\
-z_{ij} \leq Q_{ij} \leq z_{ij}, \quad \forall i, j \quad i \neq j
\end{array} \right. \]

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in SOS \quad \Rightarrow \quad Q \in S^n_+ \quad \Rightarrow \quad \text{Nonnegative Eigenvalues} \quad \Rightarrow \quad \text{SDP} \]

\[ p(x) \in DSOS \quad \Rightarrow \quad Q \in S^n_{dd} \quad \Rightarrow \quad \text{Diagonally Dominant Matrix} \]

\[ Q_{ii} \geq \sum_{i \neq j} |Q_{ij}|, \quad i = 1, \ldots, n \]

\[ \begin{align*}
Q_{ii} & \geq \sum_{i \neq j} z_{ij}, \quad i = 1, \ldots, n \\
-z_{ij} & \leq Q_{ij} \leq z_{ij}, \quad \forall i, j \ i \neq j
\end{align*} \]

Nonnegative Polynomials

\[ p(x) \geq 0 \]

\[ p(x) = B^T(x)QB(x) \]

\[ p(x) \in SOS \overset{\text{Relaxation}}{\iff} Q \in \mathcal{S}_+^n \quad \overset{\text{PSD Matrix}}{\iff} \quad \text{Nonnegative Eigenvalues} \overset{\text{Relaxation}}{\iff} \text{SDP} \]

\[ p(x) \in DSOS \overset{\text{Relaxation}}{\iff} Q \in \mathcal{S}_{dd}^n \quad \overset{\text{Diagonally Dominant Matrix}}{\iff} \quad Q_{ii} \geq \sum_{i \neq j} |Q_{ij}| \Bigg/ z_{ij}, \quad i = 1, \ldots, n \]

\[ \begin{cases} Q_{ii} \geq \sum_{i \neq j} z_{ij}, & i = 1, \ldots, n \\ -z_{ij} \leq Q_{ij} \leq z_{ij}, & \forall i, j \ i \neq j \end{cases} \]

Unconstrained optimization

\[
\begin{align*}
\text{minimize} & \quad p(x) \\
\text{subject to} & \quad x \in \mathbb{R}^n
\end{align*}
\]

maximize \[\gamma\]
subject to \[p(x) - \gamma \geq 0, \quad \forall x \in \mathbb{R}^n\]

**SOS Programming: SOS SDP**

maximize \[\gamma\]
subject to \[p(x) - \gamma = B^T(x)Q B(x)\]
\[Q \in S^n_+\]

**DSOS Programming: Linear Program**

maximize \[\gamma\]
subject to \[p(x) - \gamma = B^T(x)Q B(x)\]
\[Q \in S^n_{dd}\]
Unconstrained optimization

\[ \begin{align*}
\text{minimize} & \quad p(x) \\
\text{subject to} & \quad p(x) - \gamma \geq 0, \quad \forall x \in \mathbb{R}^n
\end{align*} \]

Constrained optimization

\[ \begin{align*}
\text{minimize} & \quad p(x) \\
\text{subject to} & \quad x \in \mathbf{K} = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \; i = 1, \ldots, m \}
\end{align*} \]

SOS Programming: SOS SDP

\[ \begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma = B^T(x)Q B(x) \\
& \quad Q \in S_+^n
\end{align*} \]

DSOS Programming: Linear Program

\[ \begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma = B^T(x)Q B(x) \\
& \quad Q \in S_{dd}^n
\end{align*} \]

SOS Programming: SOS SDP

\[ \begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0 B_d(x) \\
& \quad \sigma_i(x) = B^T_{d_i}(x)Q_i B_{d_i}(x), \; i = 1, \ldots, m \\
& \quad Q_i \in S_+^n, \; i = 0, \ldots, m
\end{align*} \]

DSOS Programming: Linear Program

\[ \begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0 B_d(x) \\
& \quad \sigma_i(x) = B^T_{d_i}(x)Q_i B_{d_i}(x), \; i = 1, \ldots, m \\
& \quad Q_i \in S_{dd}^n, \; i = 0, \ldots, m
\end{align*} \]
DSOS programming searches a small subset of nonnegative polynomials set (conservative).
- DSOS programming searches a small subset of nonnegative polynomials set (conservative).

- To improve the results, we need to increase the search space.
- For this, we define “scaled-diagonally-dominant SOS” Polynomials (SDSOS).
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is dd.
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is dd.

$$Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \notin dd$$

1 $\geq |0| + |2| \quad \times$

3 $\geq |0| + |0| \quad \checkmark$

4 $\geq |2| + |0| \quad \checkmark$
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is dd.

$$Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \notin dd$$

- $1 \geq |0| + |2| \xmark$
- $3 \geq |0| + |0| \cmark$
- $4 \geq |2| + |0| \cmark$

$$D \succcurlyeq 0 \quad Q \quad D \succcurlyeq 0$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{bmatrix} \in dd$$

- $1 \geq |0| + |1| \cmark$
- $3 \geq |0| + |0| \cmark$
- $1 \geq |0| + |1| \cmark$

$\Rightarrow Q$ is sdd.
Scaled Diagonally Dominant Matrix (sdd)

\( Q \in S^n \) is sdd, if there exists a diagonal matrix \( D \) with positive diagonal entries, such that \( DQD \) is dd.

\[
Q = \begin{bmatrix}
1 & 0 & 2 \\
0 & 3 & 0 \\
2 & 0 & 4 \\
\end{bmatrix} \notin dd
\]

\[
\begin{align*}
1 & \geq |0| + |2| & \times \\
3 & \geq |0| + |0| & \checkmark \\
4 & \geq |2| + |0| & \checkmark
\end{align*}
\]

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & \frac{1}{2} \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 2 \\
0 & 3 & 0 \\
2 & 0 & 4 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & \frac{1}{2} \\
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 1 \\
0 & 3 & 0 \\
1 & 0 & 1 \\
\end{bmatrix} \in dd
\]

\[
\begin{align*}
1 & \geq |0| + |1| & \checkmark \\
3 & \geq |0| + |0| & \checkmark \\
1 & \geq |0| + |1| & \checkmark
\end{align*}
\]

\( Q \) is sdd.

\[
S^n_{dd} \subset S^n_{sdd} \subset S^n_+
\]

Every dd matrix is sdd matrix with \( D = I \)
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \text{ is sdd if and only if it can be written as } Q = \sum_{i,j=1,...,n, i<j} M^{ij} \]

where, \( M^{ij} \in S^n \)
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \text{ is sdd if and only if it can be written as} \quad Q = \sum_{i,j=1,\ldots,n, i<j} M^{ij} \]

where, \( M^{ij} \in S^n \) with zero everywhere except at most for 4 entries

\[(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}\]

which makes the 2 \( \times \) 2 matrix

\[
\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix}
\]

symmetric and positive semidefinite.
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \] is sdd if and only if it can be written as

\[ Q = \sum_{i,j=1,...,n,i<j} M^{ij} \]

where, \( M^{ij} \in S^n \) with zero everywhere except at most for 4 entries

\[ (M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj} \]

which makes the \( 2 \times 2 \) matrix

\[ \begin{bmatrix} (M^{ij})_{ii} & (M^{ij})_{ij} \\ (M^{ij})_{ji} & (M^{ij})_{jj} \end{bmatrix} \]

symmetric and positive semidefinite.

Example: \( Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \) is not sdd, \( \notin sdd \)

\[ Q = \sum_{i,j=1,2,3,i<j} M^{ij} = M^{12} + M^{13} + M^{23} \]

\[ \begin{bmatrix} (M^{12})_{11}, (M^{12})_{12}, (M^{12})_{21}, (M^{12})_{22} \\ (M^{13})_{11}, (M^{13})_{13}, (M^{13})_{31}, (M^{13})_{33} \\ (M^{23})_{22}, (M^{23})_{23}, (M^{23})_{32}, (M^{23})_{33} \end{bmatrix} \]

\[ Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} = M^{12} + M^{13} + M^{23} \]
Scaled Diagonally Dominant Matrix (sdd)

To characterize the “sdd” matrices in terms of its element, we use the following result:

\[ Q \in S^n \text{ is sdd if and only if it can be written as } Q = \sum_{i,j=1,\ldots,n, i<j} M^{ij} \]

where, \( M^{ij} \in S^n \) with zero everywhere except at most for 4 entries

\[
(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}
\]

which makes the \( 2 \times 2 \) matrix

\[
\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix}
\]

symmetric and positive semidefinite.

Example: \( Q = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 4 \end{bmatrix} \notin dd, \in sdd \quad \rightarrow \quad Q = \sum_{i,j=1,2,3, i<j} M^{ij} = M^{12} + M^{13} + M^{23} \)
Scaled Diagonally Dominant Matrix (sdd)

\[ S^n_{dd} \subset S^n_{sdd} \subset S^n_+ \]

Every dd matrix is sdd matrix with \( D = I \)

Every sdd matrix is sum of psd matrices \( M^{ij} \)

\text{p(x) } \in \text{ SOS} \quad p(x) = B^T(x)QB(x) \quad Q \in \mathcal{S}^n_+ \quad \text{PSD Matrix} \quad \text{SDP} \\

\text{p(x) } \in \text{ DSOS} \quad p(x) = B^T(x)QB(x) \quad Q \in S^n_{dd} \quad \text{LP} \\

\text{p(x) } \in \text{ SDSOS} \quad p(x) = B^T(x)QB(x) \quad Q \in S^n_{sdd} \quad ?
Scaled Diagonally Dominant Matrix (sdd)

$Q \in \mathcal{S}^n$ is sdd if and only if it can be written as

$$Q = \sum_{i,j=1,\ldots,n, i<j} M^{ij}$$

where, $M^{ij} \in \mathcal{S}^n$ with zero everywhere except at most for 4 entries

$$(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}$$

which makes the $2 \times 2$ matrix

$$\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix}$$

symmetric and positive semidefinite.
Scaled Diagonally Dominant Matrix (sdd)

$Q \in \mathcal{S}^n$ is sdd if and only if it can be written as

$$Q = \sum_{i,j=1,\ldots,n, i < j} M^{ij}$$

where, $M^{ij} \in \mathcal{S}^n$ with zero everywhere except at most for 4 entries

$$\begin{bmatrix}
(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}
\end{bmatrix}$$

which makes the $2 \times 2$ matrix

$$\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix}$$

symmetric and positive semidefinite.

1) $(M^{ij})_{ii} + (M^{ij})_{jj} \geq 0$

2) $(M^{ij})_{ii} (M^{ij})_{jj} - (M^{ij})_{ji} (M^{ij})_{ij} \geq 0$

$\begin{bmatrix}
(M^{ij})_{ii} & (M^{ij})_{ij} \\
(M^{ij})_{ji} & (M^{ij})_{jj}
\end{bmatrix} \succeq 0$

$\text{trace}(.) = \lambda_1 + \lambda_2 \geq 0$

$\det(.) = \lambda_1 \lambda_2 \geq 0$

$\lambda_1 \geq 0, \lambda_2 \geq 0$
Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd if and only if it can be written as

$$Q = \sum_{i,j=1,\ldots,n, i<j} M^{ij}$$

where, $M^{ij} \in S^n$ with zero everywhere except at most for 4 entries

$$(M^{ij})_{ii}, (M^{ij})_{ij}, (M^{ij})_{ji}, (M^{ij})_{jj}$$

which makes the $2 \times 2$ matrix

$$\begin{bmatrix} (M^{ij})_{ii} & (M^{ij})_{ij} \\ (M^{ij})_{ji} & (M^{ij})_{jj} \end{bmatrix}$$

symmetric and positive semidefinite.

\[\begin{align*}
\text{trace}(.) &= \lambda_1 + \lambda_2 \geq 0 \\
\text{det}(.) &= \lambda_1 \lambda_2 \geq 0 \\
\lambda_1 &\geq 0, \quad \lambda_2 \geq 0
\end{align*}\]

1) $(M^{ij})_{ii} + (M^{ij})_{jj} \geq 0$

2) $(M^{ij})_{ii}(M^{ij})_{jj} - (M^{ij})_{ii}(M^{ij})_{jj} \geq 0$

$$\|C_ix + d_i\|_2 \leq e_i^T x + f_i, \quad i = 1, \ldots, m$$

Second Order Cone

$$\begin{bmatrix} 2(M^{ij})_{ii} \\ (M^{ij})_{ii} - (M^{ij})_{jj} \end{bmatrix} \leq (M^{ij})_{ii} + (M^{ij})_{jj}$$

Scaled Diagonally Dominant Matrix (sdd)

$Q \in S^n$ is sdd, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is dd.

$$S^n_{dd} \subset S^n_{sdd} \subset S^n_+$$

Every dd matrix is sdd matrix with $D = I$  
Every sdd matrix is sum of psd matrices $M^{ij}$

\[
p(x) = B^T(x)QB(x)
\]

$p(x) \in SOS$ $Q \in S^n_+$ SDP

$p(x) \in DSOS$ $Q \in S^n_{dd}$ LP

$p(x) \in SDSOS$ $Q \in S^n_{sdd}$ Second Order Cone Program (SOCP)
Scaled Diagonally Dominant Matrix (sdd)

$Q \in \mathcal{S}^n$ is sdd, if there exist a diagonal matrix $D$ with positive diagonal entries, such that $DQD$ is dd.

$$\mathcal{S}^n_{dd} \subset \mathcal{S}^n_{sdd} \subset \mathcal{S}^n_+$$

Every dd matrix is sdd matrix with $D = I$.

Every sdd matrix is sum of psd matrices $M^{ij}$.

\[
\begin{align*}
\text{p}(x) \in \text{sos} & \iff Q \in \mathcal{S}_+^n \iff \text{SDP} \\
\text{p}(x) \in \text{dsos} & \iff Q \in \mathcal{S}_{dd}^n \iff \text{LP} \\
\text{p}(x) \in \text{sdso} & \iff Q \in \mathcal{S}_{sdd}^n \iff \text{Second Order Cone Program (SOCP)}
\end{align*}
\]
**Unconstrained optimization**

\[
\text{minimize} \quad p(x) \\
\text{subject to} \quad p(x) - \gamma \geq 0, \quad \forall x \in \mathbb{R}^n
\]

**Constrained optimization**

\[
\text{minimize} \quad p(x) \\
\text{subject to} \quad x \in K = \{x \in \mathbb{R}^n : g_i(x) \geq 0, \quad i = 1, \ldots, m\}
\]

**SOS Programming: SOS SDP**

\[
\text{maximize} \quad \gamma \\
\text{subject to} \quad p(x) - \gamma = B^T(x)QB(x) \\
Q \in S^n_+
\]

**SOS Programming: SOS SDP**

\[
\text{maximize} \quad \gamma \\
\text{subject to} \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \\
\sigma_i(x) = B_d^T(x)Q_iB_d(x), \quad i = 1, \ldots, m \\
Q_i \in S^n_+, i = 0, \ldots, m
\]

**SDSOS Programming: SOCP**

\[
\text{maximize} \quad \gamma \\
\text{subject to} \quad p(x) - \gamma = B^T(x)QB(x) \\
Q \in S^n_{sd}
\]

**SDSOS Programming: SOCP**

\[
\text{maximize} \quad \gamma \\
\text{subject to} \quad p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \\
\sigma_i(x) = B_d^T(x)Q_iB_d(x), \quad i = 1, \ldots, m \\
Q_i \in S^n_{sd}, i = 0, \ldots, m
\]
SDSOS/DSOS Programming

SPOTT: MATLAB package for DSOS and SDSOS optimization written using the SPOT toolbox.


- A. Majumdar, A. A. Ahmadi, R. Tedrake, “Control and verification of high-dimensional systems with DSOS and SDSOS programming”, 53rd IEEE Conference on Decision and Control 2014

Applications:
Control and analyze of high dimensional systems
\[ P^* = \min_{x \in \mathbb{R}^2} 3 + 2x_1 + 2x_2 + 3x_1^2 + 2x_1x_2 + 3x_2^2 + x_1^4 + x_2^4 \]

**SDSOS Programming in SPOT**

```matlab
x = msspoly('x',2); % variables x1, x2
prog = spotsosprog; % DSOS/SDSOS Programming
prog = prog.withIndeterminate(x); %
p = 3+2*x(1)+2*x(2)+3*x(1)^2+2*x(1)*x(2)+3*x(2)^2+x(1)^4+x(2)^4; % p(x)
[prog, gamma] = prog.newFree(1); % variable \( \gamma \)
prog = prog . withSDSOS (p-gamma); % \( p(x) - \gamma \in DSOS/SDSOS/SOS \)
sol = prog . minimize ( -gamma,@spot_mosek); % SDP solver, solve SDSOS programming
double(sol.eval(gamma)) % obtained lower bound
```

\[ P^2_{sos} = 2.5074 = P^* \quad P^2_{sdsos} = 2.0877 \leq P^2_{sos} \quad P^2_{dsos} = 1 \leq P^2_{sdsos} \]

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_1.m
\[ P^* = \min_{x \in \mathbb{R}^2} \quad (1 + x_1 x_2)^2 - x_1 x_2 + (1 - x_2)^2 \]

subject to \( x \in K = \{ x \in \mathbb{R}^2 : 3 - 2x_2 - x_1^2 - x_2^2 \geq 0, -x_1 - x_2 - x_1x_2 \geq 0, 1 + x_1x_2 \geq 0 \} \)

**SDSOS Programming in SPOT**

\[ d = 1; \]
\[ x = msspoly('x', 2); \]
\[ prog = spotsosprog; \]
\[ prog = prog.withIndeterminate(x); \]
\[ p = (1 + x(1) * x(2))^2 - x(1) * x(2) + (1 - x(2))^2; \]
\[ g = [3 - 2 * x(2) - x(1)^2 - x(2)^2; -x(1) - x(2) - x(1)*x(2); 1 + x(1)*x(2)]; \]
\[ [prog, gamma] = prog.newFree(1); \]
\[ mos = monomials(x, 0:2*d); \]
\[ [prog, coeffs1] = prog.newFree(length(mos)); s1 = coeffs1' * mos; \]
\[ [prog, coeffs2] = prog.newFree(length(mos)); s2 = coeffs2' * mos; \]
\[ [prog, coeffs3] = prog.newFree(length(mos)); s3 = coeffs3' * mos; \]
\[ prog = prog . withSDSOS (p - gamma - [s1 s2 s3] * g); \]
\[ prog = prog . withSDSOS (s1); \]
\[ prog = prog . withSDSOS (s2); \]
\[ prog = prog . withSDSOS (s3); \]
\[ sol = prog . minimize ( -gamma, @spot_mosek); \]
\[ double(sol.eval(gamma)) \]

**SOS Polynomials**

- **SDSOS Polynomials**
- **DSOS Polynomials**
- **DSOS Polynomials**

\[ P_{\text{sos}}^* = 0.7549 = P^* \]
\[ P_{\text{dsos}}^* = 0.7549 = P_{\text{sos}}^* \]
\[ P_{\text{dsos}}^1 \leq P_{\text{dsos}}^1 \]
\[ P_{\text{dsos}}^{\text{P}2} = 0.6585 \quad P_{\text{dsos}}^{\text{P}3} = 0.6891 \quad P_{\text{dsos}}^{\text{P}4} = 0.6935 \quad P_{\text{dsos}}^{\text{P}5} = 0.6937 \quad P_{\text{dsos}}^{\text{P}6} = 0.6937 \]


60 Fall 2019
\[ \begin{align*}
\text{minimize} & \quad \frac{1}{3}x_1^6 - \frac{21}{10}x_1^4 + 4x_1^2 + x_1x_2 + 4x_2^4 - 4x_2^2 + \frac{3}{2} \\
\text{subject to} & \quad x \in K = \{ x \in \mathbb{R}^2 : -\frac{1}{16}x_1^4 + \frac{1}{4}x_1^3 - \frac{1}{4}x_1^2 - \frac{9}{100}x_2^2 + \frac{29}{400} \geq 0 \} 
\end{align*} \]

\[ P_{sos}^* = 0.4684 = P^* \]

\[ P_{sdos}^3 \leq P_{sos}^1 \quad P_{sdos}^5 = 0.3132 \quad P_{sdos}^7 = 0.3538 \]

\[ P_{dsos}^1 \leq P_{sdos}^1 \quad P_{dsos}^5 = -0.0061 \quad P_{dsos}^7 = -0.0353 \]

[https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_3.m](https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_3.m)

**Main Benefit:**
SDSOS/DSOS can scale to problems where SOS programming ceases to run due to memory/computation constraints.

Illustrative Example:

\[ P^* = \min_{x \in \mathbb{R}^n} \quad 5 + \sum_{i=1}^{n} (x_i - 1)^2 \]

\[ p^* = 5, \quad x^* = [1, 1, \ldots, 1]^T \in \mathbb{R}^n \]

Number of variables | Polynomial of order 2
--- | ---
- **SOS:** Variables:200 | Relaxation Order=1 | time= 286.5458 (s) | \( p^* = 5 \) | sdp solver: mosek
- **SDSOS:** Variables:200 | Relaxation Order=1 | time= 3.6338 (s) | \( p^* = 5 \) | sdp solver: mosek
- **DSOS:** Variables:200 | Relaxation Order=1 | time=2.6824 (s) | \( p^* = 5 \) | sdp solver: mosek

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_compare_Uncons.m
Bounded Degree SOS

Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) g_i(x) \]

\[ \sigma_i(x) \in \text{SOS}_{2d_i}, \ i = 0, \ldots, m \]

**SDP**

\[ p(x) - \sum_{i=1}^{m} \sigma_i(x) g_i(x) = B_d(x) Q_0 B_d(x) \]

\[ \sigma_i(x) = B_d(x)^T Q_i B_d(x), \ i = 1, \ldots, m \]

\[ Q_i \in \mathcal{S}_+^n, \ i = 0, \ldots, m \]

SDP Relaxation
Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) g_i(x) \]

\[ \sigma_i(x) \in SOS_{2d_i}, \ i = 0, \ldots, m \]

Krivine-Stengle’s Positivity Certificate

Let \( K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, \ldots, m \} \) (normalized polynomials)
Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x) \]
\[ \sigma_i(x) \in \text{SOS}_{2d_i}, \ i = 0, \ldots, m \]

SDP Relaxation

\[ p(x) - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \]
\[ \sigma_i(x) = B_{d_i}^T(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \]
\[ Q_i \in \mathcal{S}_+^n, i = 0, \ldots, m \]

Krivine-Stengle’s Positivity Certificate

Let \( K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, \ldots, m \} \) (normalized polynomials)

\[ p(x) = \sum_{\alpha, \beta \in \mathbb{N}^m} \lambda_{\alpha \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \]

Unknowns: \( \lambda_{\alpha \beta} \) Finitely many Nonnegative scalars

\[ \text{Theorem 2.23. Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.} \]
Nonnegative polynomial

\[ p(x) \geq 0, \, \forall x \in \mathbf{K} = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \, i = 1, \ldots, m \} \]

Putinar’s Positivity Certificate

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x) \]

\( \sigma_i(x) \in S_{2d_i}^i, \, i = 0, \ldots, m \)

SDP Relaxation

\[ p(x) - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \]

\[ \sigma_i(x) = B_{d_i}^T(x)Q_iB_{d_i}(x), \, i = 1, \ldots, m \]

Krive-Stengle’s Positivity Certificate

\[
\begin{cases} 
  x \in \mathbf{K} & p(x) = \sum \lambda_{\alpha_1}^{\alpha_1}(x)g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} + \cdots + \cdots \geq 0 \quad p(x) \geq 0 \\
  x \notin \mathbf{K} & p(x) = \sum \lambda_{\alpha_1}^{\alpha_1}(x)g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} + \cdots + \cdots \geq 0 \quad \text{or} \quad p(x) \leq 0 \\
  g_1(x) \leq 0 \text{ or } g_1(x) \geq 1
\end{cases}
\]

or

\[ p(x) \geq 0, \, \forall x \in \mathbf{K} \]

\[ \text{Theorem 2.23. Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.} \]
Nonnegative polynomial

\[ p(x) \geq 0, \quad \forall x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \quad i = 1, \ldots, m \} \]

Putinar’s Positivity Certificate

\[
p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x)
\]

\[
\sigma_i(x) \in \text{SOS}_{2d_i}, \quad i = 0, \ldots, m
\]

\[
p(x) - \sum_{i=1}^{m} \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x)
\]

\[
\sigma_i(x) = B_{d_i}^T(x)Q_iB_{d_i}(x), \quad i = 1, \ldots, m
\]

\[
Q_i \in S_+^n, \quad i = 0, \ldots, m
\]

Krivine-Stengle’s Positivity Certificate

Let \( K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \quad i = 1, \ldots, m \} \) (normalized polynomials)

\[
p(x) = \sum_{\alpha, \beta \in \mathbb{N}^m} \lambda_{\alpha, \beta} g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m}
\]

Unknowns: \( \lambda_{\alpha, \beta} \) Finitely many Nonnegative scalars

- Determining if \( p(x) \geq 0, \quad \forall x \in K \) leads to a linear optimization feasibility problem.


\[ \textbf{P}^* = \min_{x \in \mathbb{R}^n} p(x) \]
\[ \text{subject to } x \in \textbf{K} = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \]

\[ \text{maximize } \gamma \]
\[ \text{subject to } p(x) - \gamma \geq 0, \ \forall x \in \textbf{K} \]

**SDP Relaxation**

\[ \text{maximize } \gamma \]
\[ \text{subject to } p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x) g_i(x) = B_d(x)Q_0B_d(x) \]
\[ \sigma_i(x) = B_d^T(x)Q_iB_d(x), \ i = 1, \ldots, m \]
\[ Q_i \in S_+^n, i = 0, \ldots, m \]

**LP Relaxation**

\[ \textbf{P}_{L}^{*^d} = \text{maximize } \gamma \]
\[ \text{subject to } p(x) - \gamma = \sum_{\alpha, \beta \in \mathbb{N}^m} \lambda_{\alpha, \beta} g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m} \]
\[ \forall \alpha, \beta \in \mathbb{N}^m, \sum_{j=1}^{m} \alpha_j + \beta_j \leq d \]

**Theorem:** Let \( \textbf{K} \) be compact (Archimedeian). \( \textbf{P}_{L}^{*^d} \leq \textbf{P}_{L}^{*^{d+1}} \)
\[ \lim_{d \to \infty} \textbf{P}_{L}^{*^d} = \textbf{P}^* \]

**Theorem 5.10.** Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.
LP-relaxations suffer from several serious theoretical and practical drawbacks:

- The LPs of the hierarchy are numerically **ill-conditioned**.
  - It involves products of arbitrary powers of the $g_i(x)$ 's and $(1 - g_i(x))$'s.
  - In particular, the presence of large coefficients is source of ill-conditioning and numerical instability.

- The sequence of the associated optimal values converges to the global optimum only **asymptotically** and **not in finitely many steps**. (Appendix II)

- Finite convergence even does not hold for **convex optimizations**. (In standard SOS finite convergence takes place for SOS-convex problems)
LP-relaxations suffer from several serious theoretical and practical drawbacks:

- The LPs of the hierarchy are numerically ill-conditioned.
  - It involves products of arbitrary powers of the \( g_i(x) \)'s and \( (1 - g_i(x)) \)'s.
  - In particular, the presence of large coefficients is source of ill-conditioning and numerical instability.

- The sequence of the associated optimal values converges to the global optimum only asymptotically and not in finitely many steps. (Appendix II)

- Finite convergence even does not hold for convex optimizations. (In standard SOS finite convergence takes place for SOS-convex problems)

**Bounded Degree SOS (BSOS):**
Hierarchy of convex relaxations which combines some of the advantages of the SOS and LP hierarchies.
Bounded Degree SOS (BSOS):
Hierarchy of convex relaxations which combines some of the advantages of the SOS- and LP- hierarchies.

\[ p(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x) \]
\[ \sigma_0(x) \in \text{SOS}_{2d} \]
\[ \sigma_i(x) \in \text{SOS}_{2d_i}, \ i = 1, \ldots, m \]

\[ p(x) = \sum_{\substack{\alpha, \beta \in \mathbb{N}^m \\ \sum_{j=1}^{m} \alpha_j + \beta_j \leq i}} \lambda_{\alpha \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \]

\[ p(x) = \sigma_0(x) + \sum_{\substack{\alpha, \beta \in \mathbb{N}^m \\ \sum_{j=1}^{m} \alpha_j + \beta_j \leq d}} \lambda_{\alpha \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \]
\[ \sigma_0(x) \in \text{SOS}_{2k} \]

\[ P^* = \min_{x \in \mathbb{R}^n} p(x) \]

subject to \( x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m \} \)

\[ \text{maximize} \quad \gamma \]

\[ \gamma, Q_i \]

subject to \( p(x) - \gamma - \sum_{i=1}^m \sigma_i(x)g_i(x) = B_d(x)Q_0B_d(x) \)

\[ \sigma_i(x) = B_{d_i}^T(x)Q_iB_{d_i}(x), \ i = 1, \ldots, m \]

\[ Q_i \in S^n_+, i = 0, \ldots, m \]

\[ \text{maximize} \quad \gamma \]

\[ \gamma, \lambda, \alpha, \beta \geq 0, Q_0 \]

subject to \( p(x) - \gamma = \sum_{\alpha, \beta \in \mathbb{N}^m} \lambda_{\alpha, \beta} g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m} \)

\[ \forall \alpha, \beta \in \mathbb{N}^m, \sum_{j=1}^m \alpha_j + \beta_j \leq d \]

\[ Q_0 \in S^n_+ \]

**Theorem:** Let \( k \in \mathbb{N} \) be fixed.

\[ P_{d}^{*k} \leq P_{d+1}^{*k} \quad \lim_{d \to \infty} P_{d}^{*k} = P^* \]

- **Finite convergence (Like standard SOS)** (Finite convergence condition: Rank condition of the dual (moment) problem) (Appendix III)

- Unlike standard SOS, the size of SDP is fixed \( \binom{n+k}{n} \)

---

Section 1.1, Jean B. Lasserre, Kim-Chuan Toh, Shouguang Yang, "A bounded degree SOS hierarchy for polynomial optimization", EURO Journal on Computational Optimization March 2017, Volume 5, Issue 1–2, pp 87–117
Example 1

\[(P_1)\quad f = x_1^2 - x_2^2 + x_3^2 - x_4^2 + x_5 - x_2\]
\[\text{s.t.} \quad 0 \leq g_1 = 2x_1^2 + 3x_2^2 + 2x_1x_2 + 2x_3^2 + 3x_4^2 + 2x_5 + 2x_3x_4 \leq 1\]
\[0 \leq g_2 = 3x_1^2 + 2x_2^2 - 4x_1x_2 + 3x_3^2 + 2x_4^2 - 4x_3x_4 \leq 1\]
\[0 \leq g_3 = x_1^2 + 6x_2^2 - 4x_1x_2 + x_3^2 + 6x_4^2 - 4x_3x_4 \leq 1\]
\[0 \leq g_4 = x_1^2 + 4x_2^2 - 3x_1x_2 + x_3^2 + 4x_4^2 - 3x_3x_4 \leq 1\]
\[0 \leq g_5 = 2x_1^2 + 5x_2^2 + 3x_1x_2 + 2x_3^2 + 5x_4^2 + 3x_3x_4 \leq 1\]
\[0 \leq x.\]

\[P^{*}_{k=1} = P^{*}\]

Example 2

\[(P_2)\quad f = x_1^2x_2^3 + x_1^3x_2^2 - x_1^2x_2^2\]
\[\text{s.t.} \quad 0 \leq g_1 = x_1^2 + x_2^2 \leq 1\]
\[0 \leq g_2 = 3x_1^2 + 2x_2^2 - 4x_1x_2 \leq 1\]
\[0 \leq g_3 = x_1^2 + 6x_2^2 - 4x_1x_2 + 2.5 \leq 1\]
\[0 \leq g_4 = x_1^2 + 3x_2^2 \leq 1\]
\[0 \leq g_5 = x_1^2 + x_2^2 \leq 1\]
\[0 \leq x_1, \quad 0 \leq x_2.\]

\[P^{*} = -0.037037\]

Fixed size of SDP

\[k = 3\]
\[P^{*}_{k=3, d=1} = -0.041855\]
\[P^{*}_{k=3, d=2} = -0.037139\]
\[P^{*}_{k=3, d=3} = -0.037087\]
\[P^{*}_{k=3, d=4} = -0.037073\]
\[P^{*}_{k=3, d=5} = -0.037046\]

\[k = 4\]
\[P^{*}_{k=4, d=1} = -0.038596\]
\[P^{*}_{k=4, d=2} = -0.037046\]
\[P^{*}_{k=4, d=3} = -0.037040\]
\[P^{*}_{k=4, d=4} = -0.037038\]
\[P^{*}_{k=4, d=5} = -0.037037\]

More examples:  
https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Bounded_Degree_SOS/BSOS_Example1.m

https://github.com/tweisser/Sparse_BSOS/tree/master/test_suite/Dense

Code: https://github.com/tweisser/Sparse_BSOS

Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Spars Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3
Sparse SOS


- Take advantage of structure (sparsity) of the problem to solve smaller SDP
Take advantage of structure (sparsity) of the problem to solve smaller SDP

1) PSD Constraint obtained from SOS/Moment Relaxation.

- (Under some conditions) We can replace Constraint of the form $Q \succeq 0$ by PSD constraints of set of smaller matrices.

Example:

$$Q = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \Rightarrow \quad Q \text{ is PSD because:}$$

\[
\begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix} \succeq 0 \quad \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix} \succeq 0
\]
Take advantage of structure (sparsity) of the problem to solve smaller SDP

1) PSD Constraint obtained form SOS/Moment Relaxation.
   - (Under some conditions) We can replace constraint of the form $Q \succeq 0$ by PSD constraints of set of smaller matrices.

   Example:
   $$Q = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 1 & 2 \end{bmatrix} \quad \rightarrow \quad Q \text{ is PSD because:} \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix} \succeq 0 \quad \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix} \succeq 0

2) SOS relaxation of nonnegative Polynomials
   - (Under some conditions) We can replace constraint of $p(x) \in \text{SOS}$ by SOS constraints of low dimensional polynomials.

   Example:
   $$p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_2 x_3 + x_3^2) \quad \rightarrow \quad p(x_1, x_2, x_3) = p_1(x_1, x_2) + p_2(x_2, x_3)$$

   $$p_1(x_1, x_2) = (1 + x_1)^2 + (x_1 + x_2)^2$$
   $$p_2(x_2, x_3) = (1 + x_3^2)^2 + (x_2 + x_3)^2$$

   Polynomial $p(x_1, x_2, x_3)$ is SOS because $p_1(x_1, x_2)$ and $p_2(x_2, x_3)$ are SOS.
Sparse Polynomials

Polynomial: \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \quad p(x) = \sum_\alpha p_\alpha x^\alpha \)

number of coefficients \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial:** Polynomial is fully dense if all the coefficients are nonzero
Sparse Polynomials

**Polynomial:** \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \quad p(x) = \sum_{\alpha} p_{\alpha} x^{\alpha} \quad \text{number of coefficients} \quad \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial:** Polynomial is fully dense if all the coefficients are nonzero

- **Sparse polynomial:** Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

**Example: Sparse Polynomial**

\[ p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2^2 \]

Number of nonzero coefficients: 4

Number of all coefficients: \( \binom{2+5}{2} = \frac{(7)!}{2!5!} = 21 \)
Sparse Polynomials

Polynomial:  \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)  
\[ p(x) = \sum_{\alpha} p_\alpha x^\alpha \]
number of coefficients  \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial**: Polynomial is fully dense if all the coefficients are nonzero

- **Sparse polynomial**: Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

Example: Sparse Polynomial  
\[ p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2 \]
Number of nonzero coefficients: 4  
Number of all coefficients:  \( \binom{2+5}{2} = \frac{(7)!}{2!5!} = 21 \)

- **Correlative Sparsity**: It describes coupling between the variables \( x_1, \ldots, x_n \) of a polynomial  \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)
  
  - Variables \( x_i \) and \( x_j \) are coupled if they appear simultaneously in a monomial of the polynomial.
Sparse Polynomials

Polynomial: \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \quad p(x) = \sum \alpha p_\alpha x^\alpha \) number of coefficients \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial:** Polynomial is fully dense if all the coefficients are nonzero

- **Sparse polynomial:** Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

Example: Sparse Polynomial \( p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2^2 \) Number of nonzero coefficients: 4
Number of all coefficients: \( \binom{2+5}{2} = \frac{(7)!}{2!5!} = 21 \)

- **Correlative Sparsity:** It describes coupling between the variables \( x_1, \ldots, x_n \) of a polynomial \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)

  - Variables \( x_i \) and \( x_j \) are coupled if they appear simultaneously in a monomial of the polynomial.

Example: \( p(x_1, x_2, x_3, x_4) = 0.56 + 0.5x_1 + 2x_1x_2^2 + 0.75x_3^3x_4^2 \) Coupled variables: \( (x_1, x_2), (x_3, x_4) \)

  Missing Coupled variables: \( (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4) \)
**Sparse Polynomials**

Polynomial: \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)  
\[ p(x) = \sum_{\alpha} p_{\alpha} x^\alpha \]

number of coefficients  \( \binom{n+d}{n} = \frac{(n+d)!}{n!d!} \)

- **Fully dense polynomial**: Polynomial is fully dense if all the coefficients are nonzero

- **Sparse polynomial**: Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

Example: Sparse Polynomial  
\[ p(x_1, x_2) = 0.56 + 0.5x_1 + 2x_2^2 + 0.75x_1^3x_2^2 \]

Number of nonzero coefficients: 4  
Number of all coefficients: \( \binom{2+5}{2} = \frac{(7)!}{2!5!} = 21 \)

- **Correlative Sparsity**: It describes coupling between the variables \( x_1, \ldots, x_n \) of a polynomial \( p(x) : \mathbb{R}^n \rightarrow \mathbb{R} \)

  - Variables \( x_i \) and \( x_j \) are coupled if they appear simultaneously in a monomial of the polynomial.

Example:  
\[ p(x_1, x_2, x_3, x_4) = 0.56 + 0.5x_1 + 2x_1x_2^2 + 0.75x_3^3x_4^2 \]

Coupled variables: \( (x_1, x_2), (x_3, x_4) \)

Missing Coupled variables: \( (x_1, x_3), (x_1, x_4), (x_2, x_3), (x_2, x_4) \)

- Number of all possible coupling between variables \( x_1, \ldots, x_n \): \( \binom{n}{2} \)

- Polynomial has **correlative sparsity** if the number of coupled variables is much smaller than the number of all possible coupling.
Sparse Polynomials

- **Sparse polynomial**: Polynomial is sparse if the number of nonzero coefficients is much smaller than the number of the total coefficients.

- **Correlative Sparsity**: Polynomial has correlative sparsity if the number of coupled variables is much smaller than the Number of all possible coupling.
Sparse Polynomials

- **Sparse polynomial**: Polynomial is **sparse** if the number of nonzero coefficients is much smaller than the number of the total coefficients.

- **Correlative Sparsity**: Polynomial has **correlative sparsity** if the number of coupled variables is much smaller than the number of all possible coupling.

  - Correlative sparsity is a special case of the sparsity.
  - Correlative sparsity implies the sparsity, but the converse is not necessarily true.

  \[
p(x_1, x_2, x_3, x_4) = x_1x_2 + x_1x_3 + x_1^3x_4 + x_2x_3 + x_2x_4 + x_3x_4^{10}
\]

  - Sparse Polynomial
  - With NO correlative sparsity

  - Number of nonzero coefficients: 6
  - \[
  \binom{4+10}{4} = \frac{(14)!}{4!10!} = 1001
  \]
(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \quad \iff \quad p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of **low dimensional** polynomials.

\[
p(x) \in SOS \quad \iff \quad p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x)QB(x) \quad Q \in S^n_+
\]

\[
p(x) = \sum_k z_k^T(x)Q_kz_k(x) \quad Q_k \in S^{C_k}_+ \quad C_k < n
\]

(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x)QB(x) \quad Q \in S_+^n
\]

\( \iff \)

\[
p(x) = \sum_k z_k^T(x)Q_kz_k(x) \quad Q_k \in S_+^{C_k} \quad C_k < n
\]


Example:

\[
p(x_1, x_2, x_3) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_2x_3 + 2x_3^2
\]

\( p(x) \in SOS \)

\[
p(x_1, x_2, x_3) = p_1(x_1, x_2) + p_2(x_2, x_3)
\]

\[
p_1(x_1, x_2) = (\sqrt{2}x_1 + \sqrt{0.5}x_2)^2 \in SOS
\]

\[
p_2(x_2, x_3) = (\sqrt{0.5}x_2 + \sqrt{2}x_3)^2 \in SOS
\]
(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of \textbf{low dimensional} polynomials.

\[
p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x)QB(x) \quad Q \in S^n_+
\]

\( p(x) = \sum_k z_k^T(x)Q_kz_k(x) \quad Q_k \in S_{++}^{C_k} \quad C_k < n
\]

\( C_k \times C_k \) matrix \( C_k \times 1 \) monomial vector


Example:

\[
p(x_1, x_2, x_3) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_2x_3 + 2x_3^2
\]

\( p(x) \in SOS \)

\[
p(x_1, x_2, x_3) = (\sqrt{2}x_1 + \sqrt{0.5}x_2)^2 \in SOS
\]

\[
p_1(x_1, x_2) = (\sqrt{2}x_1 + \sqrt{0.5}x_2)^2 \in SOS
\]

\[
p_2(x_2, x_3) = (\sqrt{0.5}x_2 + \sqrt{2}x_3)^2 \in SOS
\]
(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \quad \iff \quad p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x)QB(x) \quad Q \in S^n_+
\]

\[
p(x) = \sum_k z_k^T(x)Q_kz_k(x) \quad Q_k \in S^{C_k}_+ \quad C_k < n
\]

(Under some conditions) Constraint of the form \( p(x) \in \text{SOS} \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in \text{SOS} \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in \text{SOS}
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x)QB(x) \quad Q \in S_+^n \iff p(x) = \sum_k z_k^T(x)Q_kz_k(x) \quad Q_k \in S_+^{C_k} \quad C_k < n
\]


(Under some conditions) Constraint of the form \( X \succeq 0 \) can be replaced by PSD constraints of smaller matrices \( X_k \succeq 0 \)

\[
X \succeq 0 \iff X = \sum_k E_k^T X_k E_k \quad X_k \succeq 0 \quad C_k < n
\]


(Under some conditions) Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS \\
X_k: \text{Coupled set variables of } p(x)
\]

\[
p(x) = B^T(x)Q B(x) \quad \text{If and only if} \quad p(x) = \sum_k z_k^T(x)Q_k z_k(x) \quad Q_k \in S^n_{++} \quad C_k < n
\]


(Under some conditions) Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$

\[
X \succeq 0 \iff X = \sum_k E_k^T X_k E_k \quad C_k < n
\]

Example:

\[
X = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad \text{If and only if} \quad X_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X_2 = \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 2 \end{bmatrix}
\]

MIT 16.S498: Risk Aware and Robust Nonlinear Planning  Fall 2019
(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

\( X_k \): Coupled set variables of \( p(x) \)

\[
p(x) = B^T(x)QB(x) \quad Q \in S^n_+
\]

\[
p(x) = \sum_k z_k^T(x)Q_k(z_k(x)) \quad Q_k \in S^C_k_+ \quad C_k < n
\]


(Under some conditions) Constraint of the form \( X \succeq 0 \) can be replaced by PSD constraints of smaller matrices \( X_k \succeq 0 \)

\[
X \succeq 0 \iff X = \sum_k E_k^T X_k E_k \quad X_k \succeq 0
\]


(Under some conditions) Constraint of the form \( p(x) \in SOS \) can be replaced by SOS constraints of low dimensional polynomials.

\[
p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS \quad X_k: \text{Coupled set variables of } p(x)
\]

\[
p(x) = B^T(x)QB(x) \quad Q \in S^n_+ \iff p(x) = \sum_k \tilde{z}_k^T(x)Q_k \tilde{z}_k(x) \quad Q_k \in S^{C_k}_+ \quad C_k < n
\]


(Under some conditions) Constraint of the form \( X \succeq 0 \) can be replaced by PSD constraints of smaller matrices \( X_k \succeq 0 \)

\[
X \succeq 0 \iff X = \sum_k E_k^T X_k E_k \quad X_k \succeq 0 \quad C_k < n
\]


Results rely on sparsity pattern of polynomials and Matrices and its graph representation, and Chordality of sparsity graph (the classical theory of graph and cliques).
**Undirected Graph**

- Undirected graph $\mathcal{G}$
- $\mathcal{V}$ Set of nodes of the graph
- $\mathcal{E}$ Set of edges of the graph
Undirected Graph  ➤ Undirected graph $G$

\[ \mathcal{V} \]  Set of nodes of the graph

\[ \mathcal{E} \]  Set of edges of the graph

- We use undirected graph to represent polynomials and symmetric matrices.

\[
p(x_1, x_2, x_3) = 1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_1 x_3 + x_2^2 + x_2 x_3 + x_3^2
\]

Coupled variables: $(x_1, x_2), (x_2, x_3)$

Edges between coupled variables

\[
X = \begin{bmatrix}
X_{11} & X_{12} & 0 & 0 \\
X_{12} & X_{22} & X_{23} & X_{24} \\
0 & X_{23} & X_{33} & X_{34} \\
0 & X_{24} & X_{34} & X_{44}
\end{bmatrix}
\]

Edges: Nonzero entries of matrix

sparsity pattern of polynomial

sparsity pattern of matrix
Undirected Graph

- Undirected graph $G$
- $\mathcal{V}$ Set of nodes of the graph
- $\mathcal{E}$ Set of edges of the graph

**Cycle:** A cycle of length $k$ in an undirected graph is a sequence of nodes $(v_1, v_2, \ldots, v_k)$ such that $(v_i, v_{i+1}) \quad i = 1, \ldots, k - 1$ and $(v_1, v_k)$ are the edges.
**Undirected Graph** ➢ Undirected graph $\mathcal{G}$ ➢ Set of nodes of the graph $\mathcal{V}$ ➢ Set of edges of the graph $\mathcal{E}$

**Cycle:** A cycle of length $k$ in a undirected graph is a sequence of nodes $(v_1, v_2, \ldots, v_k)$ such that $(v_i, v_{i+1})$ for $i = 1, \ldots, k - 1$ and $(v_1, v_k)$ are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.
**Undirected Graph**

- Undirected graph $\mathcal{G}$
  - $\mathcal{V}$ Set of nodes of the graph
  - $\mathcal{E}$ Set of edges of the graph

**Cycle:** A cycle of length $k$ in a undirected graph is a sequence of nodes $(v_1, v_2, ..., v_k)$ such that $(v_i, v_{i+1})$ $i = 1, ..., k - 1$ and $(v_1, v_k)$ are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.

**Chordal Graph:** An undirected graph is chordal if every cycle of the length $k \geq 4$ has a chord, (if there are no cycles of length $\geq 4$)
**Undirected Graph**

$\mathcal{G}$

\[ \mathcal{V} \quad \text{Set of nodes of the graph} \]

\[ \mathcal{E} \quad \text{Set of edges of the graph} \]

**Cycle:** A cycle of length $k$ in an undirected graph is a sequence of nodes $(v_1, v_2, ..., v_k)$ such that $(v_i, v_{i+1}) i = 1, ..., k - 1$ and $(v_1, v_k)$ are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.

**Chordal Graph:** An undirected graph is chordal if every cycle of the length $k \geq 4$ has a chord, (if there are no cycles of length $\geq 4$)

**Clique:** a clique of a graph is a subset of nodes that construct a complete graph (i.e. each node in the clique is connected to all the nodes in the clique.)
**Undirected Graph**  
Undirected graph $G = (V, E)$

- $V$ Set of nodes of the graph
- $E$ Set of edges of the graph

**Cycle:** A cycle of length $k$ in an undirected graph is a sequence of nodes $(v_1, v_2, ..., v_k)$ such that $(v_i, v_{i+1}) i = 1, ..., k - 1$ and $(v_1, v_k)$ are the edges.

**Chord:** is an edge that connects 2 nonadjacent nodes in a cycle.

**Chordal Graph:** An undirected graph is chordal if every cycle of the length $k \geq 4$ has a chord, (if there are no cycles of length $\geq 4$)

**Clique:** a clique of a graph is a subset of nodes that construct a complete graph (i.e. each node in the clique is connected to all the nodes in the clique.)

**Maximal Clique:** a clique is maximal if it is not a subset of another clique.
Theorem

Let $G(V,E)$ be a chordal graph\(^1\) with maximal cliques \(\{C_1, C_2, \ldots, C_t\}\). Then, Matrix $X \in S^n$ with sparsity pattern $G(V,E)$ is PSD if and only if there exist PSD matrices $X_k \in S^{|C_k|} \succeq 0$

\[
X \succeq 0 \iff X = \sum_k E^T_{C_k} X_k E_{C_k} \quad \quad X_k \in S^{|C_k|} \succeq 0 \quad |C_k| < n
\]

- Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$

Example:

\[
X = \begin{bmatrix}
2 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 2
\end{bmatrix}
\]

\[
X \succeq 0
\]

Theorem

Let $G(V, E)$ be a chordal graph\(^1\) with maximal cliques $\{C_1, C_2, \ldots, C_l\}$. Then, Matrix $X \in \mathcal{S}^n$ with sparsity pattern $G(V, E)$ is PSD if and only if there exist PSD matrices $X_k \in \mathcal{S}^{|C_k|}$ such that

$$X \succeq 0 \iff X = \sum_k E_{C_k}^T X_k E_{C_k} \quad X_k \in \mathcal{S}^{|C_k|} \succeq 0 \quad |C_k| < n$$

- Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$.

Example:

$$X = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

$$X \succeq 0$$

$$\exists \quad X_1 \in \mathcal{S}^{|C_1|} \succeq 0 \quad X_2 \in \mathcal{S}^{|C_2|} \succeq 0 \iff X \succeq 0$$

Theorem

Let $G(V, E)$ be a chordal graph\(^1\) with maximal cliques $\{C_1, C_2, ..., C_l\}$. Then, Matrix $X \in S^n$ with sparsity pattern $G(V, E)$ is PSD if and only if there exist PSD matrices $X_k \in S^{|C_k|} \succeq 0$:

$$X \succeq 0 \iff X = \sum_k E_{C_k}^T X_k E_{C_k} \quad X_k \in S^{|C_k|} \succeq 0 \quad |C_k| < n$$

- Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$.

Example:

$$X = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad X \succeq 0$$

$$\quad X_1 \succeq 0 \quad X_2 \succeq 0$$

**Theorem**

Let $G(\mathcal{V}, \mathcal{E})$ be a chordal graph\(^1\) with maximal cliques \(\{C_1, C_2, \ldots, C_l\}\). Then, Matrix $X \in \mathcal{S}^n$ with sparsity pattern $G(\mathcal{V}, \mathcal{E})$ is PSD if and only if there exist PSD matrices $X_k \in \mathcal{S}^{|C_k|} \succeq 0$.

$$
X \succeq 0 \iff X = \sum_k E_{C_k}^T X_k E_{C_k} \quad X_k \in \mathcal{S}^{|C_k|} \succeq 0 \quad |C_k| < n
$$

- Constraint of the form $X \succeq 0$ can be replaced by PSD constraints of smaller matrices $X_k \succeq 0$.

**Example:**

$$
X = \begin{bmatrix}
2 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 2
\end{bmatrix}
\quad \iff 
X_1 = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
2 & 1 \\
1 & 0.5 \\
1 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 \\
0 & 1 \\
0 & 1
\end{bmatrix} = \begin{bmatrix}
1 & 0 \\
0 & 1 \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
0.5 & 1 \\
1 & 2 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\quad X_1 \succeq 0
\quad X_2 \succeq 0
$$

\[\begin{array}{c}
\begin{array}{c}
X \succeq 0 \\
\iff \quad \begin{array}{c}
X_1 \succeq 0 \\
\quad \quad \quad \quad \downarrow \text{iff} \\
X_2 \succeq 0
\end{array}
\end{array}
\end{array}\]

\[\begin{array}{c}
\begin{array}{c}
\text{Maximal clique 1} \\
\text{Maximal clique 2}
\end{array}
\end{array}\]

**Diagram:**

- Chordal graph $G(\mathcal{V}, \mathcal{E})$ with maximal cliques $\{1, 2, 3\}$.
- Sparsity pattern of polynomial.

---

\[
\begin{align*}
\text{minimize} \quad & \mathbf{C} \otimes \mathbf{X} \\
\text{subject to} \quad & \mathbf{A}_i \otimes \mathbf{X} = \mathbf{b}_i \quad i = 1, \ldots, m. \\
& \mathbf{X} \succeq 0. \quad \mathbf{X} \in \mathcal{S}^n
\end{align*}
\]

Sparsity pattern of matrix \( \mathbf{X} \) : Chordal graph \( \mathcal{G}(\mathcal{V}, \mathcal{E}) \)

\[
\begin{align*}
\text{minimize} \quad & \mathbf{C} \otimes \mathbf{X} \\
\text{subject to} \quad & \mathbf{A}_i \otimes \left( \sum_k \mathbf{E}_{\mathcal{C}_k}^T \mathbf{X}_k \mathbf{E}_{\mathcal{C}_k} \right) = \mathbf{b}_i \quad i = 1, \ldots, m. \\
& \mathbf{X}_k \succeq 0, \ k = 1, 2, \ldots \quad \mathbf{X}_{k'} \in \mathcal{S}^{\left| \mathcal{C}_{k'} \right|}
\end{align*}
\]
Theorem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C - 2, ..., C_t\}$

Then, polynomial $p(x)$ is SOS if and only if:

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$$

$X_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints on low dimensional polynomials.
Theorem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C - 2, ..., C_l\}$. Then, polynomial $p(x)$ is SOS if and only if:

\[ p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS \]

$X_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints on low dimensional polynomials.

\[ p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_1^2 + x_2 x_3 + x_3^2) \]

Coupled variables: $(x_1, x_2), (x_2, x_3)$

Edges between coupled variables
Theorem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C_2, \ldots, C_l\}$.

Then, polynomial $p(x)$ is SOS if and only if:

$$p(x) \in \text{SOS} \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in \text{SOS}$$

$X_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in \text{SOS}$ can be replaced by SOS constraints on low dimensional polynomials.

$$p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_2^2 + x_2 x_3 + x_3^2)$$

Coupled variables: $(x_1, x_2)$, $(x_2, x_3)$

Edges between coupled variables
Theorem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C_2, \ldots, C_l\}$.

Then, polynomial $p(x)$ is SOS if and only if:

$$p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS$$

$x_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints on low dimensional polynomials.

$$p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_1^2 + x_2 x_3 + x_3^2)$$

Coupled variables: $(x_1, x_2), (x_2, x_3)$

Edges between coupled variables

$$p(x_1, x_2, x_3) \in SOS \text{ iff } p(x_1, x_2) = p_1(x_1, x_2) \in SOS + p_2(x_2, x_3) \in SOS$$

$$p(x_1, x_2, x_3) = (1 + x_1)^2 + (x_1 + x_2)^2 + (1 + x_3)^2 + (x_2 + x_3)^2$$

$\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a Chordal graph
Theorem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a chordal graph obtained from the polynomial $p(x)$ with maximal cliques $\{C_1, C - 2, \ldots, C_l\}$.

Then, polynomial $p(x)$ is SOS if and only if:

\[
p(x) \in SOS \iff p(x) = \sum_k p_k(X_k) \quad p_k(X_k) \in SOS
\]

$X_k$: Nodes in clique $C_k$

- Constraint of the form $p(x) \in SOS$ can be replaced by SOS constraints on low dimensional polynomials.

\[
p(x_1, x_2, x_3) = 2(1 + x_1 + x_3 + x_1^2 + x_1 x_2 + x_1^2 + x_2 x_3 + x_3^2)
\]

Coupled variables: $(x_1, x_2), (x_2, x_3)$

Edges between coupled variables

\[
p(x_1, x_2, x_3) \in SOS \iff p(x_1, x_2) = p_1(x_1, x_2) \in SOS + p_2(x_2, x_3) \in SOS
\]

\[
p(x_1, x_2, x_3) = (1 + x_1)^2 + (x_1 + x_2)^2 + (1 + x_3)^2 + (x_2 + x_3)^2
\]

$p(x_1, x_2, x_3) \in SOS \rightarrow p(x_1, x_2, x_3) \in SSOS$
Unconstrained optimization

$$\min_x p(x)$$

**SOS Program:**

$$\max_{Q \in S^n, \gamma} \gamma$$

subject to

$$p(x) - \gamma \in \text{SOS}$$

**SSOS Program:**

$$\max_{Q \in S^n, \gamma} \gamma$$

subject to

$$p(x) - \gamma \in \text{SSOS}$$
Unconstrained optimization

\[
\begin{align*}
\text{minimize} \quad & p(x) \\
\text{subject to} \quad & p(x) - \gamma \in \text{SOS}
\end{align*}
\]

SOS Program:

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma \in \text{SOS}
\end{align*}
\]

SSOS Program:

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma \in \text{SSOS}
\end{align*}
\]

Constrained optimization

\[
\begin{align*}
\text{minimize} \quad & x \quad p(x) \\
\text{subject to} \quad & g_i(x) \geq 0, \quad i = 1, \ldots, n
\end{align*}
\]

SOS Program:

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x)g_i(x) \in \text{SOS} \\
& \sigma_i(x) \in \text{SOS}_{2d_i}, \quad i = 0, \ldots, m
\end{align*}
\]

SSOS Program:

\[
\begin{align*}
\text{maximize} \quad & \gamma \\
\text{subject to} \quad & p(x) - \gamma - \sum_{i=1}^{m} \sigma_i(x)g_i(x) \in \text{SSOS} \\
& \sigma_i(x) \in \text{SOS}_{2d_i}, \quad i = 0, \ldots, m
\end{align*}
\]

should preserve the correlative sparsity of \( g_i \)
\[ p(x) - \gamma - \sum_{i=1}^{m} \left( \sigma_i(x) g_i(x) \right) \in SSOS \]
\[ \sigma_i(x) \in SOS_{2d_i}, \ i = 0, \ldots, m \]

- \( \sigma_i(x) \) should preserve the correlative sparsity of \( g_i(x) \)

- **Example:**
  \[ g_i(\tilde{x}) \]: is a polynomial in terms of subset of variables \( \tilde{x} \)
  \[ \sigma_i(\tilde{x}) \]: SOS polynomial in terms of variables \( \tilde{x} \)

More information:


Example: [https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_SOS/Example_SSOS_compare_Cons.m](https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_SOS/Example_SSOS_compare_Cons.m)
Sparse SOS using Yalmip

1) Copy “corrsparsey.m” to the folder of /modules/sos, and replace the original corrsparsey.m.

   https://github.com/zhengy09/sos_csp

2) Add the “ops.sos.csp = 1” to the Yalmip SOS optimization code.


sparsePOP 3.03 (MATLAB Package)

This package also provides the optimal solution $x^*$ of SSOS optimization.

https://sourceforge.net/projects/sparsepop/

Example 1: Unconstrained Optimization

\[ f_{cs}(x) = \sum_{i \in J} \left( (x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 ight) + (x_{i+1} - 2x_{i+2})^4 + 10(x_i - 10x_{i+3})^4, \]

\[ J = \{1, 3, 5, \ldots, n-3\} \]

<table>
<thead>
<tr>
<th>Number of variables</th>
<th>(Number of Clique)*(Size of the Clique)</th>
<th>cpu time (sparseSOS)</th>
<th>cpu time (SOS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>3*14</td>
<td>3.5e-7</td>
<td>0.6</td>
</tr>
<tr>
<td>40</td>
<td>3*38</td>
<td>8.4e-7</td>
<td>1.4</td>
</tr>
<tr>
<td>100</td>
<td>3*98</td>
<td>5.5e-7</td>
<td>3.8</td>
</tr>
<tr>
<td>200</td>
<td>3*198</td>
<td>3.0e-7</td>
<td>8.4</td>
</tr>
<tr>
<td>400</td>
<td>3*398</td>
<td>3.6e-7</td>
<td>19.3</td>
</tr>
</tbody>
</table>

Example 2: Unconstrained Optimization

\[ f_{BB}(x) = \sum_{i=1}^{n} \left( x_i (2 + 5x_i^2) + 1 - \sum_{j \in J_i} (1 + x_j)x_j \right)^2, \]

\[ J_i = \{ j \mid j \neq i, \max(1, i-5) \leq j \leq \min(n, i+1) \}. \]

<table>
<thead>
<tr>
<th>Broyden banded function</th>
<th>n</th>
<th>cl.str</th>
<th>( \varepsilon_{\text{obj}} )</th>
<th>sparse</th>
<th>dense</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6</td>
<td>6*1</td>
<td>8.0e-9</td>
<td>11.3</td>
<td>11.6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7*1</td>
<td>1.9e-8</td>
<td>69.5</td>
<td>69.5</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>7*2</td>
<td>2.8e-8</td>
<td>164.1</td>
<td>373.7</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>7*3</td>
<td>9.1e-8</td>
<td>240.3</td>
<td>1835.6</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>7*4</td>
<td>6.2e-8</td>
<td>348.7</td>
<td>8399.4</td>
</tr>
</tbody>
</table>

Illustrative Example:

\[
\begin{align*}
P^* & = \min_{x \in \mathbb{R}^n} \left( 5 + \sum_{i=1}^{n} (x_i - 1)^2 \right) \\
& = 5, \quad x^* = [1, 1, \ldots, 1]^T \in \mathbb{R}^n
\end{align*}
\]

<table>
<thead>
<tr>
<th>Number of variables</th>
<th>Polynomial of order 2</th>
<th>Relaxation Order=1</th>
<th>time=</th>
<th>(p^*)</th>
<th>sdpsolver: mosek</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOS: Variables:200</td>
<td></td>
<td>1</td>
<td>286.5458 (s)</td>
<td>5</td>
<td>mosek</td>
</tr>
<tr>
<td>SDSOS: Variables:200</td>
<td></td>
<td>1</td>
<td>3.6338 (s)</td>
<td>5</td>
<td>mosek</td>
</tr>
<tr>
<td>DSOS: Variables:200</td>
<td></td>
<td>1</td>
<td>2.6824 (s)</td>
<td>5</td>
<td>mosek</td>
</tr>
<tr>
<td>Spars SOS: Variables:200</td>
<td></td>
<td>1</td>
<td>0.2374 (s)</td>
<td>5</td>
<td>mosek</td>
</tr>
<tr>
<td>SparsPOP: Variables:200</td>
<td></td>
<td>1</td>
<td>0.95 (s)</td>
<td>5</td>
<td>sdpt3</td>
</tr>
</tbody>
</table>

x^* = [1, ..., 1]

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/SDSOS-DSOS/Example_SDSOS_compare_Uncons.m

https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_SOS/Example_SSOS_compare_Uncons.m
Topics

1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Spars Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3
Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)

- **Combines** Bounded degree SOS (BSOS) and Chordal-Sparse SOS.

- Takes advantages of sparsity of the original problem to reduce the size of the bounded degree SOS.

- It relies on “Running Intersection Property” (Chordal sparsity of the graph)

- Example: https://github.com/jasour/rarnop19/blob/master/Lecture6_modified-SOS/Sparse_Bounded_Degree_SOS/SBSOS_Example1.m

- MATLAB Code
  
  https://github.com/tweisser/Sparse_BSOS

  This package also provides the optimal solution $x^*$ of SBSOS optimization.
Example 1: Constrained Optimization (Chained Singular Function)

\[ f := \sum_{j \in H} \left( (x_j + 10x_{j+1})^2 + 5(x_{j+2} - x_{j+3})^2 + (x_{j+1} - 2x_{j+2})^4 + 10(x_j - x_{j+3})^4 \right) \]

\[ H := \{ 2i - 1 : i = 1, \ldots, n/2 - 1 \} \]

\[ K = \left\{ x \in \mathbb{R}^p : 1 - \sum_{i \in I_\ell} x_i \geq 0, \ell = 1, \ldots, p; x_i \geq 0, i = 1, \ldots, n \right\}. \]


Application


<table>
<thead>
<tr>
<th>Number of variables</th>
<th>Chained Singular</th>
<th>rel.</th>
<th>Sparse-BSOS</th>
<th>rk</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 500</td>
<td>d = 1</td>
<td></td>
<td>-1.4485e−02*</td>
<td>1.0</td>
<td>19.6</td>
</tr>
<tr>
<td></td>
<td>d = 2</td>
<td></td>
<td>-9.7833e−10</td>
<td>1</td>
<td>17.8</td>
</tr>
<tr>
<td>n = 600</td>
<td>d = 1</td>
<td></td>
<td>-2.7372e−03*</td>
<td>1.0</td>
<td>40.1</td>
</tr>
<tr>
<td></td>
<td>d = 2</td>
<td></td>
<td>-1.2640e−09</td>
<td>1</td>
<td>21.4</td>
</tr>
<tr>
<td>n = 700</td>
<td>d = 1</td>
<td></td>
<td>-1.7548e−03*</td>
<td>1.0</td>
<td>41.6</td>
</tr>
<tr>
<td></td>
<td>d = 2</td>
<td></td>
<td>-1.7613e−09</td>
<td>1</td>
<td>25.3</td>
</tr>
<tr>
<td>n = 800</td>
<td>d = 1</td>
<td></td>
<td>-1.9438e−03*</td>
<td>1.0</td>
<td>58.9</td>
</tr>
<tr>
<td></td>
<td>d = 2</td>
<td></td>
<td>2.1935e−09</td>
<td>1</td>
<td>29.0</td>
</tr>
<tr>
<td>n = 900</td>
<td>d = 1</td>
<td></td>
<td>-1.8924e−02*</td>
<td>1.0</td>
<td>43.5</td>
</tr>
<tr>
<td></td>
<td>d = 2</td>
<td></td>
<td>-2.6072e−09</td>
<td>1</td>
<td>33.5</td>
</tr>
<tr>
<td>n = 1000</td>
<td>d = 1</td>
<td></td>
<td>-4.4914e−02*</td>
<td>1.0</td>
<td>35.5</td>
</tr>
<tr>
<td></td>
<td>d = 2</td>
<td></td>
<td>-9.3508e−10</td>
<td>1</td>
<td>39.5</td>
</tr>
</tbody>
</table>
1) (Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)
   Modified SOS optimization that results in LP and Second order cone program

2) Bounded degree SOS (BSOS)
   Modified SOS optimization that results in smaller SDP’s.

3) Spars Sum-of-Squares Optimization (SSOS)
   Takes advantage of sparsity of the original problem to generate smaller SDP.

4) Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)
   Combination of 2 and 3
(Scaled) Diagonally Dominant Sum-of-Squares Optimization (DSOS, SDSOS)

  Code: https://github.com/anirudhamajumdar/spotless/tree/spotless_isos

Bounded Degree Sum-of-Squares Optimization (BSOS)

  Code: https://github.com/tweisser/Sparse_BSOS

Sparse Sum-of-Squares Optimization (SSOS)

  Code: https://sourceforge.net/projects/sparsepop/

  Code: https://github.com/zhengy09/sos_csp

Sparse Bounded Degree Sum-of-Squares Optimization (SBSOS)

  Code: https://github.com/tweisser/Sparse_BSOS
Appendix I: SDSOS/DSOS Polynomials
**Polynomial** \[ p(x) \in \mathbb{R}[x] \]

**Nonnegative Polynomial** \[ p(x) \geq 0 \]

### Sum-Of-Squares Polynomials

\[ p(x) \in SOS \quad \iff \quad p(x) = \sum_{i=1}^{\ell} h_i^2(x) \]

where \( h_i(x) \in \mathbb{R}[x], \quad i = 1, \ldots, \ell \)

### Diagonally-Dominant-Sum-Of-Squares Polynomials

\[ p(x) \in DSOS \]

\[ p(x) = \sum_i \alpha_i m_i^2(x) + \sum_{i,j} \beta_{ij}^+(m_i(x) + m_j(x))^2 + \sum_{i,j} \beta_{ij}^-(m_i(x) - m_j(x))^2 \]

for some nonnegative scalars \( \alpha_i, \beta_{ij}^+, \beta_{ij}^- \)

### Scaled-Diagonally-Dominant-Sum-Of-Squares Polynomials

\[ p(x) \in SDSOS \]

\[ p(x) = \sum_i \alpha_i m_i^2(x) + \sum_{i,j} (\hat{\beta}_{ij}^+ m_i(x) + \hat{\beta}_{ij}^- m_j(x))^2 + \sum_{i,j} (\hat{\beta}_{ij}^- m_i(x) - \hat{\beta}_{ij}^+ m_j(x))^2 \]

for some scalars \( \alpha_i \geq 0, \hat{\beta}_{ij}^+, \hat{\beta}_{ij}^-, \hat{\beta}_{ij}^- \)

where \( Q \in S^n_{dd} \)

where \( Q \in S^n_{sdd} \)
Appendix II: Convergence of LP Relaxation
\[ \text{P*} = \min_{x \in \mathbb{R}^n} \quad p(x) \]
subject to \[ x \in K = \{ x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, ..., m \} \]
\[ \text{P*} = \max_{\gamma \in \mathbb{R}} \quad \gamma \]
subject to \[ p(x) - \gamma \geq 0, \ \forall x \in K \quad \text{optimal solution} \quad \gamma^* = p(x^*) \]

**SDP Relaxation**

\[ p(x) - \gamma^* = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x) \quad \exists \ \gamma^* \in \mathbb{R}, \sigma_0(x) \in SOS_{2d}, \ \sigma_i(x) \in SOS_{2d_i}, i = 1, ..., m \]

if \[ \gamma^* = p(x^*) = P^* \quad p(x^*) - \gamma^* = 0 \quad \sigma_0(x^*) + \sum_{i=1}^{m} \sigma_i(x^*)g_i(x^*) = 0 \]
\[ \begin{align*}
P^* &= \text{minimize} \quad p(x) \\
\text{subject to} \quad x \in K = \{x \in \mathbb{R}^n : g_i(x) \geq 0, \ i = 1, \ldots, m\} \\
\end{align*} \]

\[ \begin{align*}
P^* &= \text{maximize} \quad \gamma \\
\text{subject to} \quad p(x) - \gamma \geq 0, \ \forall x \in K \quad \xrightarrow{\text{optimal solution}} \quad \gamma^* = p(x^*) \\
\end{align*} \]

\[ p(x) - \gamma^* = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x)g_i(x) \quad \exists \ \gamma^* \in \mathbb{R}, \sigma_0(x) \in SOS_{2d}, \sigma_i(x) \in SOS_{2d_i}, i = 1, \ldots, m \]

if \( \gamma^* = p(x^*) = P^* \) \( p(x^*) - \gamma^* = 0 \)

\[ \sigma_0(x^*) + \sum_{i=1}^{m} \sigma_i(x^*)g_i(x^*) = 0 \]

---

if \( x^* \in \text{int}K \)

\[ \sigma_0(x^*) + \sum_{i=1}^{m} \frac{\sigma_i(x^*)g_i(x^*)}{g_i(x^*)} = 0 \]

Hence, This constraint is imposed by

\[ \sigma_i(x^*) \ i = 0, \ldots, m \]

( The same situation for \( x^* \in \partial K \) )
\[ P^* = \min_x x^2 - 2x + 2 \]
subject to \[ x \in K = \{ x : x(2 - x) \geq 0 \} \]

\[ P_{sos}^* = \max_{\gamma \in \mathbb{R}, \sigma_0(x) \in \text{sOS}, \sigma_1(x) \in \text{sOS}} \gamma \]
subject to \[ x^2 - 2x + 2 - \gamma = \sigma_0(x) + \sigma_1(x)x(2 - x) \]

\[ \gamma^* = 1 \quad x^* = 1 \]

\[ \sigma_0(x) = (-0.291570596593 - 0.0571934472478x1 + 0.348740011438x1^2)^2 + (-0.956549252584 + 1.50888962843x1 - 0.552282590362x1^2)^2 \]
\[ \sigma_1(x) = (-0.653185546681 + 0.653173513801x1)^2 \]

- At \( x^* = 1 \in \text{int} \ K \)

\[ p(x^*) - \gamma^* = 0 \quad \frac{\sigma_0(x^*) + \sigma_1(x^*)x^*(2 - x^*)}{0} = 0 = 1 \]
LP Relaxation

\[
p(x) - \gamma* = \sum \lambda^*_{\alpha\beta} g_1^{\alpha_1}(x)...g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1}...(1 - g_m(x))^{\beta_m}
\]

\[
\exists \gamma* \in \mathbb{R}, \lambda^*_{\alpha\beta} \geq 0
\]

if \(\gamma* = p(x*)\),

\[
p(x*) - \gamma* = 0 \quad \sum \lambda^*_{\alpha\beta} g_1^{\alpha_1}(x*)...g_m^{\alpha_m}(x*)(1 - g_1(x*))^{\beta_1}...(1 - g_m(x*))^{\beta_m} = 0
\]

\[
P^* = \min_{x \in \mathbb{R}^n} p(x)
\]

subject to \(x \in K = \{x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, ..., m\}\)

\[
P^* = \max_{\gamma \in \mathbb{R}} \gamma
\]

subject to \(p(x) - \gamma \geq 0, \ \forall x \in K \quad \text{optimal solution} \quad \gamma* = p(x*)\)

Section 5.4.2, Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.
\[ \begin{align*}
\text{P}^* = & \min_{x \in \mathbb{R}^n} \quad p(x) \\
\text{subject to} \quad & x \in \mathbf{K} = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \; i = 1, \ldots, m \} \\
\text{P}^* = & \max_{\gamma \in \mathbb{R}} \quad \gamma \\
\text{subject to} \quad & p(x) - \gamma \geq 0, \; \forall x \in \mathbf{K} \quad \text{optimal solution} \quad \gamma^* = p(x^*)
\end{align*} \]

**LP Relaxation**

\[
p(x) - \gamma^* = \sum \lambda_{\alpha\beta}^* g_1^{\alpha_1}(x) \ldots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \ldots (1 - g_m(x))^{\beta_m} \quad \exists \; \gamma^* \in \mathbb{R}, \; \lambda_{\alpha\beta}^* \geq 0
\]

if \( \gamma^* = p(x^*) = P^* \)

\[ p(x^*) - \gamma^* = 0 \]

\[ \sum \lambda_{\alpha\beta}^* g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} = 0 \]

if \( x^* \in \text{int} \mathbf{K} \)

\[ \sum \lambda_{\alpha\beta}^* g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} > 0 \]

\[ g_i(x^*) > 0 \quad (1 - g_i(x^*)) > 0 \]

---

Section 5.4.2, Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.
\[ \text{LP Relaxation} \]

\[
P^*(x) = \min_{x \in \mathbb{R}^n} p(x)
\]
subject to \( x \in K = \{ x \in \mathbb{R}^n : 0 \leq g_i(x) \leq 1, \ i = 1, ..., m \} \)

\[
P^*(\gamma) = \max_{\gamma \in \mathbb{R}} \gamma
\]
subject to \( p(x) - \gamma \geq 0, \ \forall x \in K \) \quad \text{optimal solution} \quad \gamma^* = p(x^*)

\[
p(x) - \gamma^* = \sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x) ... g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} ... (1 - g_m(x))^{\beta_m} \]
\[ \exists \ \gamma^* \in \mathbb{R}, \ \lambda_{\alpha \beta}^* \geq 0 \]
if \( \gamma^* = p(x^*) = P^* \)

\[
p(x^*) - \gamma^* = 0 \quad \sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x^*) ... g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} ... (1 - g_m(x^*))^{\beta_m} = 0
\]
if \( x^* \in \text{int} K \)

\[
\sum \lambda_{\alpha \beta}^* g_1^{\alpha_1}(x^*) ... g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} ... (1 - g_m(x^*))^{\beta_m} > 0 \quad \text{Convergences to zero}
\]

- Hence, \( \gamma^* \) (optimal solution of the original problem) can not be attained.
- \( \text{convergence cannot be finite} \)
- \( \lim_{d \to \infty} P^*_L = P^* \)

---

\[ \text{Section 5.4.2, Jean Bernard Lasserre, “Moments, Positive Polynomials and Their Applications” Imperial College Press Optimization Series, V. 1, 2009.} \]
Example:

\[
\begin{align*}
P^* &= \text{minimize} \quad p(x) = x^2 - x \\
\text{subject to} \quad x \in K &= \{x \in \mathbb{R}^n : g_1(x) = x \geq 0, \ g_2(x) = 1 - x \geq 0\} \\
x^* &= \frac{1}{2} \in \text{int} K \\
p(x^*) &= -0.25
\end{align*}
\]

LP Relaxation

\[
P^*_L = \text{maximize} \quad \gamma \\
\text{subject to} \quad p(x) - \gamma &= \sum \lambda_{\alpha, \beta} g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x) (1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \\
\forall \alpha, \beta &\in \mathbb{N}^n \\
\sum_{j=1}^m \alpha_j + \beta_j &\leq i
\]

Slow monotone convergence to \(-0.25:\)

\[
\begin{align*}
P^*_L^2 &= -\frac{1}{3} \quad P^*_L^4 = -\frac{1}{3} \quad P^*_L^6 = -0.3 \quad P^*_L^{10} = -0.27 \quad P^*_L^{15} = -0.2695
\end{align*}
\]

Example:

\[
\begin{align*}
P^* &= \text{minimize} \quad p(x) = x - x^2 \\
\text{subject to} \quad x \in K &= \{x \in \mathbb{R}^n : g_1(x) = x \geq 0, \ g_2(x) = 1 - x \geq 0\} \\
x^* &= 0.1 \in \partial K \\
p(x^*) &= 0
\end{align*}
\]

\[
p(x) - \gamma^* = g_1(x)g_2(x) \quad \rightarrow \quad x - x^2 = x(1 - x)
\]

Some of \(g_i(x)\)'s, \((1 - g_i(x))\)'s are zero. Hence, finite convergence can take place.

Appendix III: Bounded Degree SOS
Lagrangian Perspective
To gain more insight into how the BSOS optimization works, consider the following Nonlinear optimization and its dual:

\[
P^* = \min_{x \in \mathbb{R}^n} p(x)
\]

subject to \(g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m} \geq 0, \quad \forall \sum_{j=1}^m \alpha_j + \beta_j \leq d\)

---

### Lagrange multipliers

Lagrange function

\[
L(\lambda, x) = p(x) - \sum_{j=1}^m \lambda^\alpha_j \beta_j g_1^{\alpha_1}(x) \cdots g_m^{\alpha_m}(x)(1 - g_1(x))^{\beta_1} \cdots (1 - g_m(x))^{\beta_m}
\]

subject to \(\sum_{j=1}^m \alpha_j + \beta_j \leq d\)

**Dual Optimization:**

\[
P^*_{dual} = \max_{\lambda} \min_{x \in \mathbb{R}^n} L(x, \lambda)
\]

subject to \(\lambda \geq 0\)

To solve \(\min_{x \in \mathbb{R}^n} L(x, \lambda)\), we can use SOS relaxation.

\[
\max_{\gamma} \gamma
\]

subject to \(L(x, \lambda) - \gamma \geq 0\)

---

This results in BSOS formulation

\[
\max_{\gamma, Q_0 \geq 0} \gamma
\]

subject to \(L(x, \lambda) - \gamma \in SOS_k\)
\begin{align*}
\text{maximize} & \quad \gamma \\
\text{subject to} & \quad L(x, \lambda) - \gamma \in SOS_k
\end{align*}

- For \( k = 0 \), this is results in “Krivine-Stengle’s Positivity Certificate” based LP. (brutal simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))

- For \( k > 0 \), this is results in “BSOS” relaxation. (tractable simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))
maximize \[ \gamma \]
subject to \[ L(x, \lambda) - \gamma \in SOS_k \]

• For \( k = 0 \), this is results in “Krivine-Stengle’s Positivity Certificate” based LP.
  (brutal simplification of \( \min_{x \in R^n} L(x, \lambda) \))

• For \( k > 0 \), this is results in “BSOS” relaxation.
  (tractable simplification of \( \min_{x \in R^n} L(x, \lambda) \))

➤ Hence, \( \lambda_\alpha \beta \) in LP and BSOS are approximation of the Lagrange multipliers.

➤ Based on KKT optimality condition:
\[ \lambda_\alpha \beta g_1^{\alpha_1}(x^*) \cdots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \cdots (1 - g_m(x^*))^{\beta_m} = 0 \]

➤ Hence, when finite convergence in BSOS occurs:
\[ \lambda_\alpha \beta g_1^{\alpha_1}(x^*) \cdots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \cdots (1 - g_m(x^*))^{\beta_m} = 0 \]
\[ p(x^*) - \gamma^* = 0 \]
For $k = 0$, this is results in “Krivine-Stengle’s Positivity Certificate” based LP. (brutal simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))

For $k > 0$, this is results in “BSOS” relaxation. (tractable simplification of \( \min_{x \in \mathbb{R}^n} L(x, \lambda) \))

Hence, \( \lambda_{\alpha \beta} \) in LP and BSOS are approximation of the Lagrange multipliers.

Based on KKT optimality condition:

\[
\lambda_{\alpha \beta} g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} = 0
\]

Hence, when finite convergence in BSOS occurs:

\[
\lambda_{\alpha \beta} g_1^{\alpha_1}(x^*) \ldots g_m^{\alpha_m}(x^*)(1 - g_1(x^*))^{\beta_1} \ldots (1 - g_m(x^*))^{\beta_m} = 0
\]

\( p(x^*) - \gamma^* = 0 \)


Appendix IV:
Maximal Clique and Principal Submatrix
Maximal Clique and Principal Submatrix

- Matrix $X \in \mathcal{S}^n$ with sparsity pattern defined by Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$
- $\mathcal{C}_k$ is maximal clique of graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $|\mathcal{C}_k|$ nodes.
- Define matrix $E_{\mathcal{C}_k} \in \mathbb{R}^{|\mathcal{C}_k| \times n}$ as follows:
  \[
  [E_{\mathcal{C}_k}]_{ij} = \begin{cases} 
  1, & \text{if } \mathcal{C}_k(i) = j \\
  0, & \text{otherwise}
  \end{cases}
  \]

Where $\mathcal{C}_k(i)$ is $i$-th node in $\mathcal{C}_k$

\[
E_{\mathcal{C}_1} \in \mathbb{R}^2 \times 3 \quad E_{\mathcal{C}_1} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix} \quad \text{nodes in } \mathcal{C}_1 \\
E_{\mathcal{C}_2} \in \mathbb{R}^2 \times 3 \quad E_{\mathcal{C}_2} = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \quad \text{nodes in } \mathcal{C}_2
\]

\[
X_{\mathcal{C}_1} = E_{\mathcal{C}_1} X E_{\mathcal{C}_1}^T = \begin{bmatrix} 2 & 1 \\ 1 & 0.5 \end{bmatrix}
\]

\[
X_{\mathcal{C}_2} = E_{\mathcal{C}_2} X E_{\mathcal{C}_2}^T = \begin{bmatrix} 0.5 & 1 \\ 1 & 2 \end{bmatrix}
\]

\[
X = E_{\mathcal{C}_1}^T X E_{\mathcal{C}_1} + E_{\mathcal{C}_2}^T X E_{\mathcal{C}_2}
\]