Semi-Definite Programming Problem

Asst. Prof. Dr.-Ing. Sudchai Boonto

December 1, 2023

Department of Control System and Instrumentation Engineering
King Mongkut's Unniversity of Technology Thonburi
Thailand

Objective and Motivation

Objective:

- Understand the Semi-Definite Programming (SDP)

Motivation:

- Many practical problems in operations research and combinatorial optimization can be modeled or approximated as semidefinite programming problems. In automatic control theory, SDPs are used in the context of linear matrix inequalities.

Primal Semi-Definite Programming

Let $\mathcal{S}^n$ be the space of real symmetric $n \times n$ matrices. We have

$$\operatorname{tr}(AB) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{ij}$$

where $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ are two members of $\mathcal{S}^n$.

The primal Semi-Definite Programming (SDP) is defined as

$$\begin{aligned} \text{minimize} \quad & \operatorname{tr}(CX) \\ \text{subject to} \quad & \operatorname{tr}(A_i X) = b_i \text{ for } i = 1, 2, \ldots, p \\ & X \succeq 0 \end{aligned} \tag{1}$$

- An important feature of the problem is that the variable involved is a *matrix* rather than a vector.
- The SDP is closely related to several important classes of optimization problems.

Primal Semi-Definite Programming

If matrices C and A_i for $1 \leq i \leq p$ are all diagonal matrices, i.e.,

$$
\begin{aligned}
C &= \mathrm{diag}\{c\}, \qquad A_i = \mathrm{diag}\{a_i\} \text{ where } c \in \mathbb{R}^{n\times 1} \text{ and } a_i \in \mathbb{R}^{n\times 1} \\
\mathrm{tr}(CX) &= c_1x_1 + c_2x_2 + \cdots + c_nx_n = c^Tx \\
\mathrm{tr}(A_iX) &= a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = A_ix
\end{aligned}
$$

The SDP is reduced to the standard-form LP problem

$$
\begin{aligned}
\text{minimize} \quad & c^Tx \\
\text{subject to} \quad & Ax = b \\
& x \geq 0
\end{aligned}
$$

To se the similarity between $X \succeq 0$ and $x \geq 0$, we need the concept of **convex cone**l

Dual Semi-Definite Programming

Definition 1: A convex cone $\mathcal{K}$ is a convex set such that $x \in \mathcal{K}$ implies that $\alpha x \in \mathcal{K}$ for any scalar $\alpha \geq 0$.

By definition, the set $\{X : X \in \mathbb{R}^{n\times n}, X \geq 0\}$ and $\{x : x \in \mathbb{R}^{n\times 1}, x \geq 0\}$ are convex cones.

The dual LP problem

$$\begin{aligned} \text{maximize} \quad & -b^T y \\ \text{subject to} \quad & -A^T y + s = c \\ & s \geq 0 \end{aligned}$$

The dual SDP problem

$$\begin{aligned} \text{maximize} \quad & -b^T y \\ \text{subject to} \quad & -\sum_{i=1}^{p} y_i A_i + S = C \\ & S \succeq 0 \end{aligned} \tag{2}$$

Dual Semi-Definite Programming

We assume that there exist $X \in \mathcal{S}^n, y \in \mathbb{R}^p$, and $S \in \mathcal{S}^n$ with $X \succeq 0$ and $S \succeq 0$ such that X is feasible for the primal and $\{y, S\}$ is feasible for the dual, and

$$\operatorname{tr}(CX) + b^T y = \operatorname{tr}\left(-\sum_{i=1}^{p} y_i A_i + S\right) X + b^T y = \operatorname{tr}(SX) \geq 0$$

$$\operatorname{tr}\left(S^* X^*\right) = 0 \text{ where } S \succeq 0,\ X \succeq 0$$

$$S^* = C + \sum_{i=1}^{p} y_i^* A_i \text{ and } \operatorname{tr}\left(C X^*\right) + b^T y^* = 0$$

The duality gap becomes

$$\delta\left(X, y\right) = \operatorname{tr}\left(CX\right) + b^T y, \quad X \in \mathcal{F}_p \text{ and } \{y, S\} \in \mathcal{F}_d$$

$$\mathcal{F}_p = \{X : X \succeq 0, \operatorname{tr}\left(A_i X\right) = b_i \text{ for } 1 \leq i \leq p$$

$$\mathcal{F}_d = \left\{\{y, S\} : -\sum_{i=1}^{p} y_i A_i + S = C, \quad S \succeq 0\right\}$$

Dual Semi-Definite Programming

The gap $\delta(X, y)$ is nonnegative and it is reduced to zero at the solutions X^* and S^* of the primal and dual problems, respectively.

The dual SDP problem becomes

$$\begin{aligned} \text{maximize} \quad & -b^T y \\ \text{subject to} \quad & -C - \sum_{i=1}^{p} y_i A_i \succeq 0 \end{aligned}$$

Equivalent to

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & F(x) \preceq 0 \end{aligned}$$

where $c \in \mathbb{R}^{p\times 1}$, $x \in \mathbb{R}^{p\times 1}$, and

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

with $F_i \in \mathcal{S}^n$ for $0 \leq i \leq p$.

Semi-Definite Programming

$$\begin{aligned} \text{minimize} \quad & x^T H x + p^T x \quad \text{with } H \succeq 0 \\ \text{subject to} \quad & Ax \leq b \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & \delta \\ \text{subject to} \quad & x^T H x + p^T x \leq \delta \\ & Ax \leq b \end{aligned}$$

$H \succeq 0$, we can find a matrix $\hat{H}$ such that $H = \hat{H}^T \hat{H}$, hence

$$x^T H x + p^T x \leq \delta \qquad \Rightarrow \qquad \delta - p^T x - (\hat{H}x)^T (\hat{H}x) \geq 0$$

Using Schur complement, it is

$$G(\delta, x) = \begin{bmatrix} -I_n & -\hat{H}x \\ -(\hat{H}x)^T & -\delta + p^T x \end{bmatrix} \preceq 0$$

$G(\delta, x)$ is affine wiht respect to variables x and δ.

Semi-Definite Programming

The linear constraints $Ax \le b$ can be expressed as

$$F(x) = F_0 + \sum_{i=1}^{n} x_i F_i \succeq 0 \text{ where } F_0 = -\operatorname{diag}\{b\}, \quad F_i = \operatorname{diag}\{a_i\}$$

Setting $\hat{x} = \begin{bmatrix} \delta \\ x \end{bmatrix}^T$, the general convex QP problem can be reformulated as the SDP problem

$$\begin{aligned} \text{minimize} \quad & \hat{c}^T \hat{x} \\ \text{subject to} \quad & E(\hat{x}) \preceq 0 \end{aligned}$$

where $\hat{c} \in \mathbb{R}^{n+1}$ with $\hat{c} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ and

$$E(\hat{x}) = \operatorname{diag}\{G(\delta, x), F(x)\}$$

Semi-Definite Programming

The KKT conditions for the SDP (1) can be stated as follows:

- Matrix X^* is a minimizer of the SDP Problem (1) if and only if there exist a matrix $S^* \in \S^n$ and a vector $y^* \in \mathbb{R}^p$ such that

$$
\begin{aligned}
-\sum_{i=1}^{p} y_i^* A_i + S^* &= C \\
\mathrm{tr}(A_i X^*) &= b_i \qquad \text{for } 1 \leq i \leq p \\
\mathrm{tr}(S^* X^*) &= 0 \\
X^* \succeq 0, &\quad S^* \succeq 0
\end{aligned}
\tag{3}
$$

- A set $\{X^*, y^*, S^*\}$ satisfying (3) is called a **primal-dual solution**. Itt follows that $\{X^*, y^*, S^*\}$ is a primal-dual solution if an only if X^* solves the primal problem and $\{y^*,\ S^*\}$ solves the dual problem.

The central path consists of a set $\{X(\tau),\ y(\tau),\ S(\tau)\}$ such that for each $\tau > 0$ the equations satisfy

$$\begin{aligned} -\sum_{i=1}^{p} y_i(\tau)A_i + S(\tau) &= C \\ \mathrm{tr}(A_i X(\tau)) &= b_i \qquad \text{for } 1 \le i \le p \\ \mathrm{tr}(X(\tau)S(\tau)) &= \tau I \\ X(\tau) \succeq 0, \quad S(\tau) &\succeq 0 \end{aligned} \tag{4}$$

The duality gap on the central path

$$\begin{aligned} \delta\left[X(\tau), y(\tau)\right] &= \mathrm{tr}\left(CX(\tau)\right) + b^T y(\tau) = \mathrm{tr}\left(\left[-\sum_{i=1}^{p} y_i(\tau)A_i + S(\tau)\right] X(\tau)\right) + b^T y(\tau) \\ &= \mathrm{tr}\left(S(\tau)X(\tau)\right) = \mathrm{tr}(\tau I) = n\tau \quad \Rightarrow \quad \lim_{\tau \to 0} \delta\left[X(\tau), y(\tau)\right] = 0 \end{aligned}$$

Semi-Definite Programming

- The SDP usually generates iterates by obtaining approximate solution (4) for a sequence of decreasing $\tau_k > 0$ for $k = 0, 1, \ldots$. If we let

$$G(X, y, S) = \begin{bmatrix} -\sum_{i=1}^{p} y_i A_i + S - C \\ \mathrm{tr}(A_1 X) - b_1 \\ \vdots \\ \mathrm{tr}(A_p X) - b_p \\ XS - \tau I \end{bmatrix}$$

  Then the first three equation of (4) can be expressed as $G(X, y, S) = 0$.
- $X(\tau)S(\tau) = \tau I$ is rewritten in symmetric form as

$$XS + SX = 2\tau I$$

Semi-Definite Programming

- We start with a given set $\{X, y, S\}$ and find increments $\Delta X, \Delta y$, and ΔS with ΔX and ΔS symmetric such that set $\{\Delta X, \Delta y, \Delta X\}$ satisfies the linearized equations

$$\begin{aligned} -\sum_{i=1}^{p} \Delta y_i A_i + \Delta S = C - S + \sum_{i=1}^{p} y_i A_i \\ \operatorname{tr}(A_i \Delta X) = b_i - \operatorname{tr}(A_i X) \quad \text{for } 1 \leq i \leq p \\ X\Delta S + \Delta S X + \Delta X S + S \Delta X = 2\tau I - XS - SX \end{aligned} \tag{5}$$

- The equation (5) can be reformulated as

$$J \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_d \\ r_p \\ r_c \end{bmatrix}, \text{ where } J = \begin{bmatrix} 0 & -A^T & I \\ A & 0 & 0 \\ E & 0 & F \end{bmatrix} \tag{6}$$

The solution of (6) is given by

$$\begin{aligned} \Delta x &= -E^{-1}\left[F(r_d + A^T\Delta y) - r_c\right] \\ \Delta s &= r_d + A^T\Delta y \\ M\Delta y &= r_p + AE^{-1}(Fr_d - r_c) \end{aligned} \tag{7}$$

where the matrix M, which is known as the Schur complement matrix, is given by

$$M = AE^{-1}FA^T$$

Semi-Definite Programming

1. Input A_i for $1 \le i \le p$, $b \in \mathbb{R}^p$, $C \in \mathbb{R}^{n \times n}$, and a strictly feasible set $\{X_p, y_0, S_0\}$ that satisfies (1) and (2) with $X_0 \succ 0$ and $S_0 \succ 0$. Choose a scalar σ in the range $0 \le \sigma < 1$. Set $k = 0$ and initialize the tolerance ε for the duality gap δ_k.
2. Compute $\delta = \frac{\mathrm{tr}(X_k S_k)}{n}$
3. If $\delta_k \le \varepsilon$, output solution $\{X_k, y_k, S_k\}$ and stop; otherwise, set $\tau_k = \sigma \frac{\mathrm{tr}(X_k S_k)}{n}$ and continuous with Step 4
4. Solve (6) with (7) where $X = X_k$, $y = y_k$, $S = S_k$, and $\tau = \tau_k$. Convert the solution $\{\Delta x, \Delta y, \Delta s\}$ into $\{\Delta X, \Delta y, \Delta S\}$ with $\Delta X =$ mat(Δx) and $\Delta S =$ mat(Δs).
5. Choose a parameter γ in the range $0 < \gamma < 1$ and determine parameters α and β are

$$\alpha = \min(1, \gamma\hat{\alpha}, \quad \beta = \min(1, \gamma\hat{\beta})$$

Semi-Definite Programming

5. Cont. where $\hat{\alpha} = \max_{X_k+\bar{\alpha}\Delta X \succeq 0}(\bar{\alpha})$ and $\hat{\beta} = \max_{S_k+\bar{\beta}\Delta S \preceq}(\bar{\beta})$
6. Set

$$X_{k+1} = X_k + \alpha \Delta X$$
$$y_{k+1} = y_k + \beta \Delta y$$
$$S_{k+1} = S_k + \beta \Delta S$$

   Set $k = k + 1$ and repeat form Step 2.

Writing your own code is not a good idea, we will use a well test available package like CVX or JuMP instead.

Eigenvalue Problem Example

- Poles location plays very important role in the closed-loop control system design. We can obtain the poles of the linear system by solving the eigenvalues of the system matrix.
- If M is a square $n \times n$ matrix, then λ is an eigenvalue of M with corresponding eigenvector x if

$$Mx = \lambda x \quad \text{and} \quad x \neq 0$$

  The λ is an eigenvalue of M if and only if λ is a root of the polynomial:

$$p(\lambda) = \det(M - \lambda I) \text{ that is } p(\lambda) = \det(M - \lambda I) = 0$$

- If M is symmetric, then all eigenvalues λ of M must be real numbers, and these eigenvalues can be ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.

Eigenvalue Problem Example

- The corresponding eigenvectors $q^1, \ldots, q^n$ of M cna be chosen so that they are orthogonal, namely $(q^i)^T(q^j) = 0$ for $i \neq j$, and can be scaled so that $(q^i)^T(q^i) = 1$. This mean the matrix Q satisfies:

$$Q^T Q = I, \text{ and } Q^T = Q^{-1}$$

  We call it as a orthonormal matrix.

- The matrix D is

$$D = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}, \qquad M = QDQ^T$$

Eigenvalue Problem Example

- $M \succeq 0$ if an only if $M = QDQ^T$ where the eigenvalues (i.e., the diagonal entries of D) are all nonnegative.
- If $M \succeq tI$ if and only if $\lambda_{\min}(M) \geq t$. To see this, let us consider the eigenvalue decomposition of $M = QDQ^T$, and consider the matrix R defined as:

$$R = M - tI = QDQ^T - tI = Q(D - tI)Q^T$$

Then

$$M \succeq tI \iff R \succeq 0 \iff D - tI \succeq 0 \iff \lambda_{\min}(M) \geq t.$$

The last property is because D is a diagonal matrix.

Semi-Definite Programming : Example

Find scalars α_1, α_2, and α_3 such that the maximum eigenvalue of $F = A_0 + \alpha_1 A_1 + \alpha_2 A_2 + \alpha_3 A_3$ is minimized where

$$A_0 = \begin{bmatrix} 2 & -0.5 & -0.6 \\ -0.5 & 2 & 0.4 \\ -0.6 & 0.4 & 3 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, A_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Solution: As matrix F is symmetric, there exists an orthogonal matrix U such that $U^T F U = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3\}$ with $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Hence we can write

$$U^T(tI - F)U = tI - U^T F U = \mathrm{diag}(t - \lambda_1, t - \lambda_2, t - \lambda_3)$$

Semi-Definite Programming : Example

The problem can be formulated as

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & tI - F \succeq 0 \end{aligned}$$

Using CVX, we have $t = 3.000$, which is the minimized maximum eigenvalue of F.

```
clear all
A0 = [2 -0.5 -0.6;
         -0.5 2 0.4;
         -0.6 0.4 3];
A1 = [0 1 0; 1 0 0; 0 0 0];
A2 = [0 0 1; 0 0 0; 1 0 0];
A3 = [0 0 0; 0 0 1; 0 1 0];

cvx_solver sdpt3
```

```
cvx_begin
  variables y(3), t;

  F = A0 + y(1)*A1 + y(2)*A2 ...
           + y(3)*A3;
  minimize t
  subject to
        t*eye(3) - F >= 0
cvx_end
```

Semi-Definite Programming : Example

By using JuMP we obtain the same solution.

```
using JuMP,LinearAlgebra, SCS

begin
  A0 = [2 -0.5 -0.6;
       -0.5 2 0.4;
       -0.6 0.4 3];
  A1 = [0 1 0; 1 0 0; 0 0 0];
  A2 = [0 0 1; 0 0 0; 1 0 0];
  A3 = [0 0 0; 0 0 1; 0 1 0];

I = Matrix{Float64}(
     LinearAlgebra.I, 3, 3)
```

```
model = m = Model(SCS.Optimizer)
    @variable(model, y[1:3])
    @variable(model, t)
    @objective(model, Min, t)

    F = A0 + y[1]*A1 +
         y[2]*A2 + y[3]*A3
    @constraint(model,
          t .* I - F .>= 0)

    optimize!(model)
    r = objective_value(model)
    println(r)
end
```

Constraints on the $\mathcal{H}_\infty$ Norm

Consider the system with transfer function $T(s)$ as state space realization

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bw(t), \qquad x(0) = 0 \\ z(t) &= Cx(t) + Dw(t) \end{aligned}$$

Assuming that $T(s)$ is stable, the $\mathcal{H}_\infty$ norm of the system is

$$\|T\|^2\infty = \max_{w \neq 0} \frac{\int_0^\infty z^T(t) z(t) dt}{\int_0^\infty w^T(t) w(t) dt}, \quad x(0) = 0.$$

It follows that $\|T\|_\infty < \gamma$ is equivalent to

$$\int_0^\infty (z^T(t) z(t) - \gamma^2 w^T(t) w(t)) dt < 0$$

Holding true for all square integrable, non-zero $w(t)$.

Constraints on the $\mathcal{H}_\infty$ Norm

Introduce a Lyapunov function $V(x) = x^T P x$ with $P = P^T > 0$. Since $x(0) = x(\infty) = 0$, the constraint $\|T\|_\infty < \gamma$ is enforced by the existence of a matrix $P = P^T > 0$ such that

$$\frac{dV(x)}{dt} + \frac{1}{\gamma} z^T(t) z(t) - \gamma w^T(t) w(t) < 0$$

for all $x(t), w(t)$; to turn into a LMI, substitute

$$\frac{dV(x)}{dt} = x^T(A^T P + PA)x + x^T PBw + w^T B^T Px, \quad z = Cx + Dw$$

To obtain

$$\begin{bmatrix} x \\ w \end{bmatrix}^T \begin{bmatrix} A^T P + PA + \frac{1}{\gamma} C^T C & PB + \frac{1}{\gamma} C^T D \\ B^T P + \frac{1}{\gamma} D^T C & -\gamma I + \frac{1}{\gamma} D^T D \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} < 0$$

Constraints on the $\mathcal{H}_\infty$ Norm

For $\|T\|_\infty < \gamma$ the above must hold for all x and w, thus the block matrix must be negative definite. The condition can be rewritten as

$$\begin{bmatrix} A^T P + PA & PB \\ B^T P & -\gamma I \end{bmatrix} + \frac{1}{\gamma} \begin{bmatrix} C^T \\ D^T \end{bmatrix} \begin{bmatrix} C & D \end{bmatrix} < 0$$

By Schur complement, we have

Theorem (Bound real lemma)

$\|T\|_\infty < \gamma$ if and only if there exists a positive definite, symmetric matrix P that satisfies the linear matrix inequality

$$\begin{bmatrix} A^T P + PA & PB & C^T \\ B^T P & -\gamma I & D^T \\ C & D & -\gamma I \end{bmatrix} < 0$$

Constraints on the $\mathcal{H}_\infty$ Norm

```
sys = rss(3,3);
A = sys.a; B = sys.b; C = sys.c; D = sys.d;
n = size(A,1); nu = size(B,2); ny = size(D,1);

cvx_begin sdp
    variable P(n,n) symmetric
    variable gm;

    minimize gm;
    subject to
        P >= 0;
        [A'*P + P*A, P*B, C';
           B'*P , -gm*eye(nu), D';
           C, D, -gm*eye(ny)] <= 0;
cvx_end

display(P);
```

Reference

1. Joaquim R. R. A. Martins, Andrew Ning, "*Engineering Design Optimization*," Cambridge University Press, 2021.
2. Mykel J. kochenderfer, and Tim A. Wheeler, "*Algorithms for Optimization*," The MIT Press, 2019.
3. Ashok D. Belegundu, Tirupathi R. Chandrupatla, "*Optimization Concepts and Applications in Engineering*," Cambridge University Press, 2019.
4. Laurent Lessarn , "*Introduction to Optimization*," Lecture Note, University of Wisconsin–Madison.
5. Andreas Antoniou and Wu-Sheng Lu, "*Practical Optimization: Algorithms and Engineering Applications*," 2^{nd} edition, Springer, 2021.
6. Stephen Boyd an dLieven Vandenberghe "*Convex Optimization*," Cambridge University Press, 2004.