

Overview

CSDP is a solver for semidefinite programming (SDP) problems. It was originally written in 1997 and its C implementation continues to be periodically supported by the original author, Brian Borchers. Using the ASCII SDPA format¹, SDPs can be easily passed to the solver. Alternatively, problems can be directly encoded via its C API.

The purpose of this document is to give examples of how to reformulate SDPs into the form that CSDP accepts (either SDPA or its C interface). This often requires algebraic manipulation to write the problem in the primal SDP form as specified in the CSDP User Guide². Note that CSDP can work with matrices that have both symmetric blocks and diagonal blocks. The use of diagonal blocks are particularly useful for encoding and solving linear programs (LP).

Semidefinite programming is a subfield of convex optimization. These problems are defined as optimizing a linear objective function with respect to linear constraints over the space of real symmetric $n \times n$ matrices, \mathbb{S}^n . CSDP solves semidefinite programming primal–dual problems of the form

$$\begin{aligned} \mathcal{P} : \quad & \max_{X \in \mathbb{S}^n} \quad \text{tr}(CX) \\ & \text{s.t.} \quad \text{tr}(A_i X) = b_i, \forall i \in [m] \\ & \quad \quad X \succeq 0 \\ \\ \mathcal{D} : \quad & \min_{\substack{y \in \mathbb{R}^m \\ Z \in \mathbb{S}^n}} \quad b^\top y \\ & \text{s.t.} \quad \sum_{i=1}^m y_i A_i - C = Z \\ & \quad \quad Z \succeq 0, \end{aligned}$$

where $C, A_1, \dots, A_m \in \mathbb{S}^n$, $y, b \in \mathbb{R}^m$. Note that different packages and presentations of primal–dual SDP problems may differ slightly; for example, the min and the max may be swapped, or the constraint in the dual may not have the explicit decision variable Z and instead write $\sum_{i=1}^m y_i A_i - C \succeq 0$.

To put SDPs in context with other convex optimizations, note the following hierarchy³

$$\text{LP} \subseteq \text{QP} \subseteq \text{QCQP} \subseteq \text{SOCP} \subseteq \text{SDP}.$$

Linear Matrix Inequalities

A linear matrix inequality (LMI) in variable $y \in \mathbb{R}^m$ has the form⁴

$$F(y) = F_0 + y_1 F_1 + \dots + y_m F_m \succeq 0, \tag{1}$$

where $F_0, \dots, F_m \in \mathbb{S}^n$, and specifies a convex constraint on y . Many inequalities can be represented as LMIs, for example, for $d_i, z \in \mathbb{R}^N$ and $f \in \mathbb{R}^M$

$$d_i^\top z \leq f_i, \forall i \in [M],$$

can be expressed as the LMI

$$\begin{bmatrix} f_1 - d_1^\top z & 0 & \dots & 0 \\ 0 & f_2 - d_2^\top z & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f_M - d_M^\top z \end{bmatrix} \succeq 0.$$

Similarly, if we wish to express the equality constraints

$$d_i^\top z = f_i, \forall i \in [M],$$

with variable $z \in \mathbb{R}^N$ as an LMI, we could split the constraints into the pairs of inequalities

$$d_i^\top z \leq f_i, \quad d_i^\top z \geq f_i, \quad \forall i \in [M],$$

and use the same LMI form from above

$$\text{diag}([f_1 - d_1^\top z \quad \dots \quad f_M - d_M^\top z \quad d_1^\top z - f_1 \quad \dots \quad d_M^\top z - f_M]) \succeq 0.$$

¹http://plato.asu.edu/ftp/sdpa_format.txt

²<https://github.com/coin-or/Csdp/files/2485526/csdpuser.pdf>

³Princeton ORF 523 Notes

⁴EE363 Notes

Example

Let $z \in \mathbb{R}^2$ with data

$$D = \begin{bmatrix} 0 & 4 \\ 1 & -1 \\ 3 & 2 \end{bmatrix} \in \mathbb{R}^{3 \times 2}$$

$$f = [8 \quad 7 \quad 9]^\top \in \mathbb{R}^3.$$

We can extract the rows of D as $D^\top = [d_1^\top \quad d_2^\top \quad d_3^\top] \in \mathbb{R}^{2 \times 3}$. Writing out the constraints

$$\begin{array}{rcl} 4z_2 \leq 8 & & 8 - 4z_2 \geq 0 \\ z_1 - z_2 \leq 7 & \iff & 7 - z_1 + z_2 \geq 0, \\ 3z_1 + 2z_2 \leq 9 & & 9 - 3z_1 - 2z_2 \geq 0 \end{array}$$

it is clear to see that an LMI can be written as above.

Reformulating an LP as an SDP

Any LP is a special instance of an SDP⁵, in particular its dual (as defined in this document). Given the following LP

$$\begin{array}{ll} \min_{y \in \mathbb{R}^m} & b^\top y \\ \text{s.t.} & Ay + c \geq 0, \end{array} \quad (2)$$

where $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$, and $c \in \mathbb{R}^n$, it can be written as the dual SDP

$$\begin{array}{ll} \min_{y \in \mathbb{R}^m} & b^\top y \\ \text{s.t.} & F(y) \succeq 0, \end{array} \quad (3)$$

where $F(y) = F_0 + y_1 F_1 + \cdots + y_m F_m$ is an LMI. Given that $A = [a_1 \quad \dots \quad a_m] \in \mathbb{R}^{n \times m}$, the definitions of F_i are as follows

$$F_0 = \text{diag}(c) \in \mathbb{S}^n$$

$$F_i = \text{diag}(a_i) \in \mathbb{S}^n, \quad \forall i \in [m].$$

Note that

$$F(y) = \text{diag}(\{c_j + y_1 a_{1j} + \cdots + y_m a_{mj}\}_{j=1}^n) \quad (4)$$

$$= \text{diag}(c^\top + y^\top A^\top) = \text{diag}(Ay + c). \quad (5)$$

Example

Consider the following LP⁶, with $m = 3$ and $n = 10$

$$\begin{array}{ll} \max_{p,q,r \in \mathbb{R}} & z = 45p + 60q + 50r \\ \text{s.t.} & 20p + 10q + 10r \leq 2400 \\ & 12p + 28q + 16r \leq 2400 \\ & 15p + 6q + 16r \leq 2400 \\ & 20p + 15q \leq 2400 \\ & 0 \leq p \leq 100 \\ & 0 \leq q \leq 40 \\ & 0 \leq r \leq 60 \end{array} \quad (6)$$

where the optimal solution is $p^* = 81.82$, $q^* = 16.36$, and $r^* = 60$. The optimal objective value is $z^* = 7664$.

⁵Slide 13 CMU 10-725

⁶Slide 16 of MIT OCW 15.053 Lecture Notes.

It turns out that an LP is really a special case of an SDP where all the matrices are diagonal (positive semidefiniteness for a diagonal matrix means nonnegativity of its diagonal elements). Our goal is to formulate this as an SDP so that we can use the CSDP solver. To do this, we observe that the LP (particularly, the objective) is most like the dual part of this primal–dual SDP pair (i.e., the primary decision variable of the dual is $y \in \mathbb{R}^n$, which is like our LP).

First, we massage the constraints of (6) into a form that looks more like the dual SDP

$$\begin{aligned}
\min_{p,q,r \in \mathbb{R}} \quad & z = 45p + 60q + 50r \\
\text{s.t.} \quad & 20p + 10q + 10r + 2400 \geq 0 \\
& 12p + 28q + 16r + 2400 \geq 0 \\
& 15p + 6q + 16r + 2400 \geq 0 \\
& 20p + 15q + 2400 \geq 0 \\
& p + 100 \geq 0 \\
& q + 40 \geq 0 \\
& r + 60 \geq 0 \\
& -p + 0 \geq 0 \\
& -q + 0 \geq 0 \\
& -r + 0 \geq 0
\end{aligned} \tag{7}$$

Note that we did three things: (1) rearranged the inequalities so that 0 is on the RHS; (2) flipped the sign of the objective so that the problem is a min. Simultaneously, we redefined p , q , and r to be their negative. This way, the resulting objective has plus signs and each term of the constraints (except the last three constraints) are negative; (3) flipped the sign of the inequalities by multiplying both sides by -1 .

By pattern matching, we make the following assignments

$$\begin{aligned}
y &= [p \quad q \quad r]^\top \in \mathbb{R}^m; \quad b = [40 \quad 60 \quad 50]^\top \in \mathbb{R}^m \\
C &= -\text{diag}([2400 \quad 2400 \quad 2400 \quad 2400 \quad 100 \quad 40 \quad 60 \quad 0 \quad 0 \quad 0]) \in \mathbb{S}^n \\
A_1 &= \text{diag}([20 \quad 12 \quad 15 \quad 20 \quad 1 \quad 0 \quad 0 \quad -1 \quad 0 \quad 0]) \in \mathbb{S}^n \\
A_2 &= \text{diag}([10 \quad 28 \quad 6 \quad 15 \quad 0 \quad 1 \quad 0 \quad 0 \quad -1 \quad 0]) \in \mathbb{S}^n \\
A_3 &= \text{diag}([10 \quad 16 \quad 16 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad -1]) \in \mathbb{S}^n.
\end{aligned}$$

Thus, the following primal problem that encodes LP (6) can be input into CSDP⁷

$$\begin{aligned}
\max_{X \in \mathbb{S}^n} \quad & \text{tr}(CX) \\
\text{s.t.} \quad & \text{tr}(A_1 X) = b_1 = 45 \\
& \text{tr}(A_2 X) = b_2 = 60 \\
& \text{tr}(A_3 X) = b_3 = 50 \\
& X \succeq 0.
\end{aligned} \tag{8}$$

Note that CSDP provides the optimal values of X , y , and Z . The values we are interested in are in y , because we reformulated LP (6) as the dual.

Eigenvalue Optimization

Consider the task of minimizing the maximum eigenvalue of a symmetric matrix. This is useful in many cases, including optimizing for the characteristics of dynamical systems. Using the LMI $A(y)$, the problem can be stated as

$$\min_{y \in \mathbb{R}^m} \quad \lambda_{\max}(A(y)).$$

Writing this as an SDP involves the epigraph formulation, where we introduce a new decision variable, $t \in \mathbb{R}$

$$\begin{aligned}
\min_{y, t} \quad & t \\
\text{s.t.} \quad & \lambda_{\max}(A(y)) \leq t
\end{aligned}
\iff
\begin{aligned}
\min_{y, t} \quad & t \\
\text{s.t.} \quad & A(y) \preceq t
\end{aligned}$$

By minimizing t , the max eigenvalue of $A(y)$ is constrained to decrease. Observe that we can use the language of semidefinite matrices to express the constraint as an SDP.

⁷lp.cpp

Example⁸

We can use an LMI to formulate the following max-eig minimization as an SDP

$$\min_{y \in \mathbb{R}^2} \lambda_{\max} \left(\begin{bmatrix} 1+y_1 & y_2-3 \\ y_2-3 & -y+1+2y_2 \end{bmatrix} \right) \iff \begin{array}{ll} \min_{y, t} & t \\ \text{s.t.} & \begin{bmatrix} 1 & -3 \\ -3 & 0 \end{bmatrix} + y_1 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + y_2 \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \preceq tI_2 \end{array} .$$

Note that we can go all the way down to the standard dual form

$$\begin{array}{ll} \min_{y \in \mathbb{R}^m} & b^\top y \\ \text{s.t.} & \sum_{i=1}^m y_i A_i - C = Z, \\ & Z \succeq 0 \end{array}$$

by augmenting the decision variable vector $y = [y_1 \ y_2 \ t]^\top \in \mathbb{R}^3$, flipping the p.s.d constraint direction, and making the following definitions

$$b = [0 \ 0 \ 1]^\top$$

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{S}^2 \quad A_2 = \begin{bmatrix} 0 & -1 \\ -1 & -2 \end{bmatrix} \in \mathbb{S}^2 \quad A_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{S}^2 \quad C = \begin{bmatrix} 1 & -3 \\ -3 & 0 \end{bmatrix} \in \mathbb{S}^2.$$

For the sake of completeness, we give the SPDA sparse ASCII format of this problem, which can be read into CSDP via the command line⁹.

```
"max-eig minimization problem
3 =n
1 =num blocks
2 =size of individual blocks
0 0 1 =bvec
0 1 1 1 1
0 1 1 2 -3
1 1 1 1 -1
1 1 2 2 1
2 1 1 2 -1
2 1 2 2 -2
3 1 1 1 1
3 1 2 2 1
```

⁸Slide 12 Purdue ECE695 Lecture

⁹As opposed to above, which can be entered using the CSDP C API.