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 format[1], 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 Guide[2]. 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

$$\begin{aligned}
\mathcal{P}: \quad & \max_{X \in \mathbb{S}^n} \quad \mathrm{tr}(CX) \\
& \text{s.t.} \quad \mathrm{tr}(A_i X) = b_i, \ \forall i \in [m] \\
& \qquad\quad X \succeq 0
\end{aligned}$$

$$\begin{aligned}
\mathcal{D}: \quad & \min_{\substack{y \in \mathbb{R}^n \\ Z \in \mathbb{S}^n}} \quad b^\top y \\
& \text{s.t.} \quad \sum_{i=1}^m y_i A_i - C = Z \\
& \qquad\quad Z \succeq 0,
\end{aligned}$$

where $C, A_1, \dots, A_m \in \mathbb{S}^n,\ y, b \in \mathbb{R}^3$. 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 $\sum_i y_i A_i - C \succeq 0$.

To put SDPs in context with other convex optimizations, note the following hierarchy[3]

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

Linear Matrix Inequalities

Reformulating an LP as an SDP

Any LP is a special instance of an SDP[4], in particular its dual (as defined in this document). Given the following LP

$$\begin{aligned}
\min_{y \in \mathbb{R}^n} \quad & b^\top y \\
\text{s.t.} \quad & Ay + c \geq 0,
\end{aligned} \tag{1}$$

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

$$\begin{aligned}
\min_{y \in \mathbb{R}^n} \quad & b^\top y \\
\text{s.t.} \quad & F(y) \succeq 0,
\end{aligned} \tag{2}$$

where $F(y) = F_0 + y_1 F_1 + \cdots + y_n F_n$ is an LMI. Given that $A = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} \in \mathbb{R}^{m \times n}$, the definitions of F_i are as follows

$$\begin{aligned}
F_0 &= \mathrm{diag}(c) \\
F_i &= \mathrm{diag}(a_i), \ \forall i \in [n].
\end{aligned}$$

Note that

$$F(y) = \mathrm{diag}(\{c_j + y_1 a_{1j} + \cdots + y_n a_{nj}\}_{j=1}^m) \tag{3}$$
$$= \mathrm{diag}(c^\top + y^\top A^\top) = \mathrm{diag}(Ay + c). \tag{4}$$

[1] `http://plato.asu.edu/ftp/sdpa_format.txt`
[2] `https://github.com/coin-or/Csdp/files/2485526/csdpuser.pdf`
[3] Princeton `ORF 523 Notes`
[4] Slide 13 `CMU 10-725`

Example

Consider the following LP[5]

$$
\begin{aligned}
\max_{p,q,r\in\mathbb{R}} \quad & z = 45p + 60q + 50r \\
\text{s.t.} \quad & 20p + 10q + 10r \leq 2400 \\
& 12p + 28q + 16r \leq 2400 \\
& 15p + \ \ 6q + 16r \leq 2400 \\
& 20p + 15q \qquad\quad \leq 2400 \\
& \qquad\quad 0 \leq p \leq 100 \\
& \qquad\quad 0 \leq q \leq 40 \\
& \qquad\quad 0 \leq r \leq 60
\end{aligned}
\tag{5}
$$

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

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 (5) into a form that looks more like the dual SDP

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

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

$$
\begin{aligned}
y &= \begin{bmatrix} p & q & r \end{bmatrix}^\top; \qquad b = \begin{bmatrix} 40 & 60 & 50 \end{bmatrix}^\top \\
C &= -\operatorname{diag}(\begin{bmatrix} 2400 & 2400 & 2400 & 2400 & 100 & 40 & 60 & 0 & 0 & 0 \end{bmatrix}) \\
A_1 &= \operatorname{diag}(\begin{bmatrix} 20 & 12 & 15 & 20 & 1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}) \\
A_2 &= \operatorname{diag}(\begin{bmatrix} 10 & 28 & 6 & 15 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix}) \\
A_2 &= \operatorname{diag}(\begin{bmatrix} 10 & 16 & 16 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}).
\end{aligned}
$$

Thus, the following primal problem that encodes LP (5) can be input into CSDP[6]

$$
\begin{aligned}
\max_{X\in\mathbb{S}^n} \quad & \operatorname{tr}(CX) \\
\text{s.t.} \quad & \operatorname{tr}(A_1 X) = b_1 = 45 \\
& \operatorname{tr}(A_2 X) = b_2 = 60 \\
& \operatorname{tr}(A_2 X) = b_3 = 50 \\
& X \succeq 0.
\end{aligned}
\tag{7}
$$

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 (5) as the dual.

[5] Slide 16 of `MIT OCW 15.053 Lecure Notes`.

[6] `lp.cpp`