Infeasible and weakly infeasible semidefinite programs in [1]

Gábor Pataki*

July 11, 2017

Abstract

This is a short and nontechnical description of the infeasible and weakly infeasible semidefinite programing (SDP) instances from [1].

1 Infeasible and weakly infeasible SDPs

We denote by $\mathcal{S}^n$ the set of $n \times n$ symmetric matrices and by $\mathcal{S}^n_+$ the set of $n \times n$ symmetric positive semidefinite (psd) matrices. Given $A_1, \dots, A_m \in \mathcal{S}^n$ we consider the linear operator

$$\mathcal{A} : \mathbb{R}^m \rightarrow \mathcal{S}^n, \text{ given as } \mathcal{A}x = \sum_{i=1}^m x_i A_i, \text{where } x \in \mathbb{R}^m,$$

and its adjoint

$$\mathcal{A}^* : \mathcal{S}^n \rightarrow \mathbb{R}^m \text{ given as } \mathcal{A}^* Y = (A_1 \bullet Y, \dots, A_m \bullet Y)^T, \text{ where } Y \in \mathcal{S}^n,$$

where the inner product $A \bullet B$ is the trace of AB.

Consider now the semidefinite system

$$\begin{array}{rcl} \mathcal{A}^* Y & = & c \\ Y & \succeq & 0 \end{array} \tag{D}$$

which we call a *dual SDP* (this is for convenience, following the convention of [1]).

We say that (D) is *infeasible*, if there is no Y that satisfies its constraints. We say it is *strongly infeasible*, if the distance of the affine subspace

$$\{ Y \,| \mathcal{A}^* Y = c \}$$

to $\mathcal{S}^n_+$ is positive; and it is *weakly infeasible*, if it is infeasible, but not strongly so.

A separation theorem from convex analysis implies that (D) is strongly infeasible, if and only there is $x \in \mathbb{R}^m$ such that

$$\mathcal{A}x \succeq 0, \, \langle c, x \rangle = -1. \tag{1.1}$$

*Department of Statistics and Operations Research, University of North Carolina at Chapel Hill

Example 1. *The semidefinite system*

$$\begin{array}{rcl} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \bullet Y & = & 0, \\ \begin{pmatrix} 0 & 1 \\ 1 & \alpha \end{pmatrix} \bullet Y & = & -1, \\ Y & \succeq & 0 \end{array} \tag{1.2}$$

is infeasible iff $\alpha \geq 0$, and weakly infeasible iff $\alpha = 0$. Indeed, if $Y = (y_{ij}) \succeq 0$ satisfies the first constraint, then $y_{11} = 0$, hence by psdness $y_{12} = 0$. We can directly check that (1.1) is feasible iff $\alpha > 0$.

The algorithms of [1] generate instances whose infeasibility and weak infeasibility are easy to verify by inspection. The following two claims explain their structure:

Claim 1. *Suppose $k \geq 1$, and $p_1, \dots, p_k, p_{k+1} \geq 0$ are integers. Also suppose that A_i is of the form*

$$A_i = \left(\begin{array}{ccc} \overbrace{\qquad\qquad}^{p_1 + \ldots + p_{i-1}} & \overbrace{\quad}^{p_i} & \overbrace{\qquad\qquad}^{n - p_1 - \ldots - p_i} \\ \times & \times & \times \\ \times & I & 0 \\ \times & 0 & 0 \end{array} \right)$$

for $i = 1, \dots, k+1$, where the $\times$ symbols correspond to blocks with arbitrary elements, $A_{k+2}, \dots, A_m$ are arbitrary and

$$c^T = (0, \dots, 0, -1, c_{k+2}, \dots, c_m).$$

Then (D) is infeasible.

Proof Suppose Y is feasible in (D). Since $A_1 \bullet Y = 0$, the upper left p_1 by p_1 block of Y is zero, and $Y \succeq 0$ proves that the first p_1 rows and columns of Y are zero. Inductively, from the first k constraints we deduce that the first $\sum_{i=1}^k p_i$ rows and columns of Y are zero.

Deleting the first $\sum_{i=1}^k p_i$ rows and columns from A_{k+1} we obtain a psd matrix, hence

$$A_{k+1} \bullet Y \quad \geq \quad 0,$$

contradicting the $(k+1)^{st}$ constraint in (D). $\square$

Claim 2. *Suppose $\ell \geq 1$ and $q_1, \dots, q_\ell, q_{\ell+1} \geq 0$ are integers. Also suppose there exist $Y_j \in \mathcal{S}^n$ of the form*

$$Y_j = \left(\begin{array}{ccc} \overbrace{\qquad\qquad}^{n - q_1 - \ldots - q_j} & \overbrace{\quad}^{q_j} & \overbrace{\qquad\qquad}^{q_1 + \ldots + q_{j-1}} \\ 0 & 0 & \times \\ 0 & I & \times \\ \times & \times & \times \end{array} \right)$$

where $j = 1, \dots, \ell+1$ and again the $\times$ symbols correspond to blocks with arbitrary elements. Also assume

$$\begin{array}{rcl} \mathcal{A}^* Y_j & = & 0 \ (j = 1, \dots, \ell) \\ \mathcal{A}^* Y_{\ell+1} & = & c. \end{array}$$

Then (D) is not strongly infeasible.

Proof Suppose (D) is strongly infeasible and let us fix $x \in \mathbb{R}^m$ to satisfy (1.1). Observe

$$\mathcal{A}x \bullet Y_j \;=\; \langle x, \mathcal{A}^* Y_j \rangle \;=\; 0 \, (j = 1, \dots, \ell),$$

hence an argument like in the proof of Claim 1 shows the last $q_1 + \cdots + q_\ell$ rows and columns of $\mathcal{A}x$ are zero. Thus

$$\begin{aligned} \langle c, x \rangle &= \langle \mathcal{A}^* Y_{\ell+1}, x \rangle \\ &= Y_{\ell+1} \bullet \mathcal{A}x \;\geq\; 0, \end{aligned}$$

a contradiction. □

Example 1 continued *If* $\alpha = 1$ *then (1.2) fits the framework of Claim 1 with* $k = 1,\, p_1 = p_2 = 1$.

If $\alpha = 0$, *then it fits the same framework with* $p_1 = 1,\, p_2 = 0$. *In this case we can choose* $\ell = 1,\, q_1 = 1,\, q_2 = 0$ *and*

$$Y_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},\, Y_2 = \begin{pmatrix} 0 & -1/2 \\ -1/2 & 0 \end{pmatrix}$$

as in Claim 2 to prove that (1.2) is not strongly infeasible.

Algorithm 2 in [1] generates infeasible SDPs with the structure given in Claim 1 with

$$n = 10,\, k = 2,\, p_1 = 2,\, p_2 = 3,\, p_3 = 2,\, m = 10 \text{ or } m = 20.$$

We call these instances *infeasible:* they may be strongly or weakly infeasible.

Algorithm 3 in [1] generates weakly infeasible SDPs, together with the Y_j of the form given in Claim 2, with

$$\begin{aligned} &n = 10,\, k = 2,\, \ell = 1,\, p_1 = 2,\, p_2 = 3,\, p_3 = 2,\, q_1 = 2,\, q_2 = 1 \\ &m = 10 \text{ or } m = 20. \end{aligned}$$

We call these instances *weakly infeasible:* these are guaranteed to be weakly infeasible.

We also add an optional

> **Messing step:** Choose $T = (t_{ij}) \in \mathbb{Z}^{m \times m}$ and $V = (v_{ij}) \in \mathbb{Z}^{n \times n}$ random invertible matrices with entries in $[-2, 2]$ and let
>
> $$A_i \;=\; V^T \Big(\sum_{j=1}^{m} t_{ij} A_j \Big) V \text{ for } i = 1, \dots, m.$$

These operations do not change the status of (D): they keep it weakly infeasible, if it was weakly infeasible; and strongly infeasible, if it was strongly infeasible.

We call the instances to which we did *not* apply the Messing step, *clean*; and the instances to which we did apply it, *messy.*

We store the instances in Sedumi format, so the roles of $\mathcal{A}$ and $\mathcal{A}^*$ are exchanged, and the right hand side is called b (not c). Furthermore, we provide I_{10} as objective function. All entries in our instances are integers and all entries in the Y_j matrices are rationals with small denominators (for details, see [1]). Thus one can verify the infeasibility and weak infeasibility of our instances in exact arithmetic (following the proofs of Claims 1 and 2).

References

[1] Minghui Liu and Gábor Pataki. Exact duals and short certificates of infeasibilty and weak infeasibility in conic linear programming. *Math. Program. A*, to appear, 2017.