SDP - Benchmark Problems

Michael Garstka

April 19, 2018

In the following a set of benchmark problems is presented. These convex optimization problems all involve positive semidefinite constraints and occur in various fields, such as portfolio optimization, graph theory, robust control, or polynomial optimization. These problems were collected to benchmark conic solvers against each other and to study the performance gains of various extensions to our conic ADMM solver. Our solver takes problems of the following format:

$$\begin{aligned} \min_{x,s} \quad & \frac{1}{2}x^T P x + q^T x \\ \text{s.t.} \quad & Ax + s = b, \\ & s \in \mathcal{K}, \end{aligned} \tag{1}$$

with decision variables $x \in \mathbb{R}^n$, $s \in \mathbb{R}^m$ and data matrices $P \in \mathcal{S}_+^n$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$. The convex cone $\mathcal{K}$ is a composition of the zero cone, the non-negative orthant, a set of second order cones, and a set of positive semidefinite cones. All test problems are available under `https://github.com/migarstka/SDP_Benchmark_Problems`. The following list gives an overview over the considered problems:

Contents

1 Random SDP with Quadratic Objective

The following random SDP with quadratic objective is considered:

$$\begin{aligned} \min_{x,s} \quad & \frac{1}{2}x^T P x + q^T x \\ \text{s.t.} \quad & Ax + s = b, \\ & \mathbf{mat}(s) \in \mathcal{S}_+, \end{aligned} \tag{2}$$

with decision variables $x \in \mathbb{R}^n$, $s \in \mathbb{R}^m$ and data matrices $P \in \mathcal{S}_+^n$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$ The dual of (2) is given by:

$$\begin{aligned}
\max_{x,s,y} \quad & -\frac{1}{2}x^T P x - b^T y + \inf_{s \in \mathcal{K}} \{y^T s\} \\
\text{s.t.} \quad & Px + A^T y = -q, \\
& y \in \mathcal{K}^*,
\end{aligned} \tag{3}$$

with dual variable $y \in \mathbb{R}^m$. The test problems are generated in the following way. n matrices $A_i \in \mathbb{R}^{r \times r}$ are randomly generated with $A_{ij} \sim \mathcal{N}(0,1)$ with 40% non-zero entries. Those matrices are vectorized and form the columns of A:

$$A = \begin{bmatrix} \text{vec}(A_1) & \dots & \text{vec}(A_n) \end{bmatrix}. \tag{4}$$

Next, feasible primal and dual points are generated: $x_f \in \mathbb{R}^n \sim \mathcal{N}(0,1)$, $S_f \in \mathbb{S}_+^n \sim \mathcal{N}(0,1)$, $Y_f \in \mathbb{S}_+^n \sim \mathcal{N}(0,1)$. P, q and b are chosen to fulfil the constraints in (2) and (3). $P \in \mathcal{S}_+^{n^2} \sim \mathcal{N}(0,1)$ is randomly chosen and q and b are given by:

$$q = -Px_f - A^T \text{vec}(Y_f), \tag{5}$$
$$b = Ax_f + \text{vec}(S_f). \tag{6}$$

2 Nearest Correlation Matrix

Consider the problem of projecting the decision variable onto the set of correlation matrices, i.e. real symmetric positive semidefinite matrices with ones on the diagonal. This problem occurs for example in Portfolio Optimization when the data matrix only approximates the correlations of stocks and is therefore not exactly a correlation matrix. However, further analysis methods might require the data matrix to belong to the set of correlation matrices. Consequently, one is interested in finding the nearest correlation matrix X to a given data matrix $C \in \mathbb{R}^{n \times n}$. This problem can be formulated as:

$$\begin{aligned}
\min_{X} \quad & \frac{1}{2} \|X - C\|_F^2 \\
\text{s.t.} \quad & X_{ii} = 1, \quad i = 1, \dots, n, \\
& X \in \mathcal{S}_+^n,
\end{aligned} \tag{7}$$

with Frobenius norm $||A||_F = \sqrt{\sum_{ij} |a_{ij}|^2}$. According to [HM11] problems like this can be efficiently solved up to $n = 5000$ on a standard computer. In order to transform the problem into the solver format, C and X are vectorized and the squared norm is expanded:

$$\begin{aligned}
\min_{x,s} \quad & \frac{1}{2}(x^T x - 2c^T x + c^T c) \\
\text{s.t.} \quad & \underbrace{\begin{bmatrix} E \\ -I \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}}_{s} = \underbrace{\begin{bmatrix} 1_{n \times 1} \\ 0_{n^2 \times 1} \end{bmatrix}}_{b} \\
& s_1 \in \{0\}, \quad s_2 \in \mathcal{S}_+^n,
\end{aligned} \tag{8}$$

with $c = \text{vec}(C) \in \mathbb{R}^{n^2}$ and $x = \text{vec}(X) \in \mathbb{R}^{n^2}$ as the stacked columns of the matrices C and X. $E \in \mathbb{R}^{n \times n^2}$ is a matrix that extracts the n diagonal entries X_{ii} form the vectorized form x.

For the benchmark problems the random data matrix C is generated as a correlation matrix using the vine method described in [LKJ09] and then perturbed by adding a random symmetric matrix $N \in \mathcal{S}^n$ with $N_{ij} \sim \mathcal{N}(0, 0.1)$ to it.

3 Smallest Sphere around multiple Ellipsoids

According to [VB96] the problem of finding the smallest sphere around a set of ellipsoids can be expressed as a semidefinite program. Consider k ellipsoids $\epsilon_1, \dots, \epsilon_k$ defined as the sublevel sets of quadratic functions of the form:

$$f_i(x) = x^T A_i x^T + 2b_i^T x + c_i, \quad i = 1, \dots, k, \tag{9}$$

i.e.

$$\epsilon_i = \{x \mid f_i(x) \leq 0\}. \tag{10}$$

The constraint that one ellipsoid ϵ contains another ellipsoid $\bar{\epsilon}$ can be posed as a LMI constraint using the $\mathcal{S}$-procedure. ϵ contains $\bar{\epsilon}$ if and only if one can find a $\tau \geq 0$ such that:

$$\begin{bmatrix} A & b \\ b^T & c \end{bmatrix} \preceq \tau \begin{bmatrix} \bar{A} & \bar{b} \\ \bar{b}^T & \bar{c} \end{bmatrix}. \tag{11}$$

A sphere can be represented as the sublevel set of the following quadratic function $f(x) = x^T I x - 2x_c^T x + \gamma \leq 0$. Consequently, it contains the set of k ellipsoids if and only if the following LMIs hold:

$$\begin{bmatrix} I & -x_c \\ -x_c^T & \gamma \end{bmatrix} \preceq \tau_i \begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix}, \quad \tau_i \geq 0, \quad i = 1, \dots, k. \tag{12}$$

The objective to minimize the radius r of the sphere can be expressed by another LMI that bounds the radius. If $r = \sqrt{x_c^T x_c - \gamma}$ then the constraint $r^2 \leq t$ is equivalently represented by:

$$\begin{bmatrix} I & x_c \\ x_c^T & \gamma + t \end{bmatrix} \succeq 0. \tag{13}$$

The problem of finding the smallest sphere around a set of ellipsoids is then given by:

$$\begin{aligned} \min_{t, xc, \gamma, \tau_1, \dots, \tau_k} \quad & t \\ \text{s.t.} \quad & \begin{bmatrix} I & -x_c \\ -x_c^T & \gamma \end{bmatrix} \preceq \tau_i \begin{bmatrix} A_i & b_i \\ b_i^T & c_i \end{bmatrix}, \quad i = 1, \dots, k. \\ & \tau_i \geq 0, \quad i = 1, \dots, k \\ & \begin{bmatrix} I & x_c \\ x_c^T & \gamma + t \end{bmatrix} \succeq 0. \end{aligned} \tag{14}$$

Figure 1 shows the solution for a problem with five randomly generated ellipsoids. The ran-

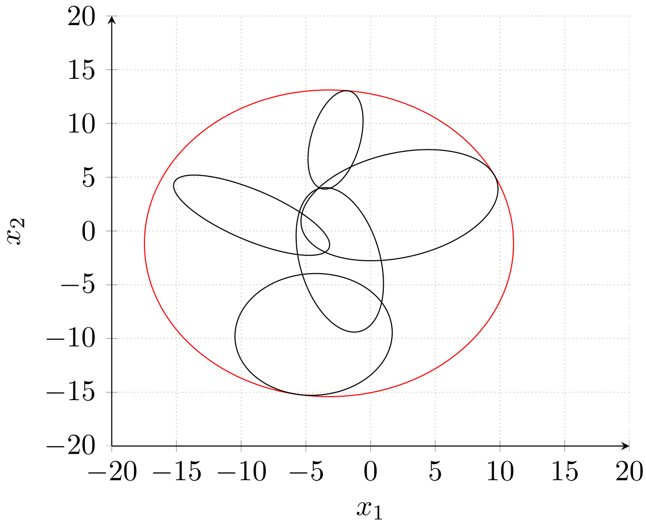

Figure 1: Smallest circle (red) around a set of randomly generated ellipsoids (black).

domly generated test problems contain between two and ten ellipsoids of dimension 2. The quadratic function representation of the ellipsoids is obtained by first generating the ellipsoids as the transformed image of a unit ball:

$$\epsilon = \{Pz + x_c \mid ||z||_2 \leq 1\}, \tag{15}$$

with $P \in \mathcal{S}^n_{++}$. The eigenvalues of P, i.e. the scaling factors of the semi-axes, are randomly chosen from the interval $[0.25, 8]$. Furthermore a rotation with a random angle $\theta \in [-\pi, \pi)$ is applied to P. The centre point x_c of each ellipsoid is randomly chosen from $-10 \leq [x_c^1, x_c^2]^T \leq 10$. Lastly, the representation (15) is turned into the quadratic function representation (9) in the following way:

$$A = P^{-2}, \quad b = -P^{-2}x_c, \quad c = x_c^T P^{-2} x_c - 1. \tag{16}$$

4 LMI-based Robust Control Problems

A wide range of robust control problems can be solved using semidefinite programming since design requirements on the controller can be specified using LMIs. Consider the closed loop system depicted in Fig 2.

The generalized plant $P(s)$ with state $x(t) \in \mathbb{R}^n$, control input $u(t) \in \mathbb{R}^{n_u}$, external input vector $w(t) \in \mathbb{R}^{n_w}$, fictitious output vector $z(t) \in \mathbb{R}^{n_z}$, and measured plant signals $v(t) \in \mathbb{R}^{n_v}$, that is connected to a controller $K(s)$. The plant dynamics are given by:

$$\dot{x}(t) = Ax(t) + B_w w(t) + B_u u(t) \tag{17}$$
$$z(t) = C_z x(t) + D_{zw} w(t) + D_{zu} u(t) \tag{18}$$
$$v(t) = C_v x(t) + D_{vw} w(t). \tag{19}$$

The problem of finding a state feedback controller of the form $u(t) = Kx(t)$ is considered, i.e. it is assumed that $C_v = I_n, D_{vw} = 0$. To find a stabilizing controller gain K, it suffices to solve

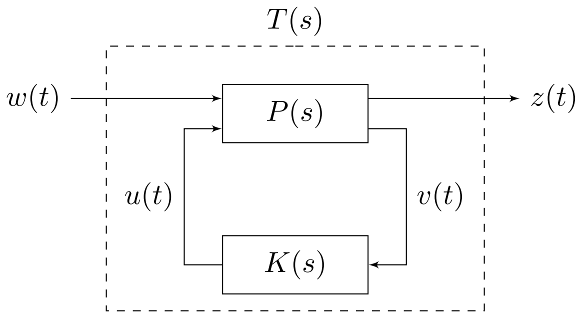

Figure 2: Generalized plant $P(s)$ connected to controller $K(s)$.

the Lyapunov inequality for the closed loop system, i.e. find a $P = P^T > 0$ and Y, s.t.

$$PA^T + AP + Y^T B_u^T + B_u Y \prec 0, \tag{20}$$

where Y is a slack variable introduced to make the inequality linear in the decision variables. The controller gain K can then be recovered from $K = YP^{-1}$.

Furthermore, it is often of interest to find a K that not only stabilizes the system but also fulfils further design requirements. Those design goals can be modelled by designing the generalized plant in such a way that minimizing a norm on the transfer function of the closed-loop system $T(s)$ achieves a desired performance. According to [GA94] a necessary and sufficient condition for a state feedback controller to achieve a H_∞-norm less than γ is the existence of matrices $P = P^T > 0$ and Y that satisfy

$$\begin{bmatrix} AP + PA^T + B_u Y + Y^T B_u^T & B_w & PC_z^T + Y^T D_{zu}^T \\ B_w^T & -\gamma I & D_{zw}^T \\ C_z P + D_{zu} Y & D_{zw} & -\gamma I \end{bmatrix} \prec 0. \tag{21}$$

In order to turn (20) and (21) into the constraint format compatible with our solver:

$$\bar{A}x + s = b, \quad s \in \mathcal{K},$$

the following vectorization rules for the product of two matrices $A \in \mathbb{R}^{k \times l}, B \in \mathbb{R}^{l \times m}$ can be used:

$$\begin{aligned} \mathbf{vec}(AB) &= (I_m \otimes A)\mathbf{vec}(B) \\ &= (B^T \otimes I_k)\mathbf{vec}(A). \end{aligned} \tag{22}$$

Thus, the constraint matrix $\bar{A}$ can be found by first computing the kroenecker products for all matrix products with P or Y inside the LMI and then adding and reordering the rows of those factors in the right way.

The test problems are created with random dimension $n, n_u, n_w, n_z \in [3, 6]$. Furthermore, a random system matrix $A \in \mathbb{R}^{n \times n}$ with $A_{ij} \sim \mathcal{N}(0, 1)$ is chosen and $B_u \in \mathbb{R}^{n \times n_u}$ is randomly taken from the set of matrices such that the controllability matrix has full rank. The remaining system matrices are selected to be $B_w = I_{n_w \times n}$, $D_{zu} = I_{n_z \times n_u}$ and random matrices $C_z \in \mathbb{R}^{n_z \times n}$ with $C_{z,ij} \sim \mathcal{N}(0, 1)$ and $D_{zw} \in \mathbb{R}^{n_z \times n_w}$ with $D_{zw,ij} \sim \mathcal{N}(0, 1)$.

5 Semidefinite Relaxation of MIQO Problems

According to[PB17] semidefinite programming can be used to approximately solve the problem of minimizing a convex quadratic function over the integer lattice $\mathbb{Z}^n$. The NP-hard problem is given by:

$$\begin{aligned} \min_{x} \quad & x^T P x + 2q^T x \\ \text{s.t.} \quad & x \in \mathbb{Z}^n, \end{aligned} \tag{23}$$

with integer variable $x \in \mathbb{Z}^n$, $P \in \mathcal{S}^n_+$, and $q \in \mathbb{R}^n$. Problems that can be transformed into (23) are for example the *integer least squares problem* ($\min_x ||Ax-b||_2^2$, s.t. $x \in \mathbb{Z}^n$) that arises in wireless communication or the *closest vector problem* ($\min_x ||v-x||_2^2$, s.t. $x \in \{Bz, z \in \mathbb{Z}^n\}$) that has applications in encryption. Two relaxation steps are introduced to turn (23) into a semidefinite program. The first step involves the relaxation of the integer constraint into a set of continuous but non-convex quadratic constraints:

$$\begin{aligned} \min_{x} \quad & x^T P x + 2q^T x \\ \text{s.t.} \quad & x_i(x_i - 1) \geq 0, \quad i = 1, \dots, n. \end{aligned} \tag{24}$$

In the second step a new matrix variable $X = xx^T$ is introduced. Now (24) can be rewritten in the following way:

$$\begin{aligned} \min_{x,X} \quad & \mathbf{Tr}(PX) + 2q^T x \\ \text{s.t.} \quad & \mathbf{diag}(X) \geq x, \\ & X = xx^T. \end{aligned} \tag{25}$$

The rewritten non-convex constraint $X = xx^T$ is then relaxed into $X \succeq xx^T$ and turned into a LMI using the Schur complement:

$$\begin{aligned} \min_{x,X} \quad & \mathbf{Tr}(PX) + 2q^T x \\ \text{s.t.} \quad & \mathbf{diag}(X) \geq x, \\ & \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0. \end{aligned} \tag{26}$$

The optimal value p^r of the relaxed problem (26) yields a lower bound on the optimal objective p^* of the original problem (23).

A number of random integer least squares test problems is created in the following way. $A \in \mathbb{R}^{m \times n}$ is generated with $A_{ij} \sim \mathcal{N}(0,1)$ and $n = \frac{1}{2}m \in [5, 15]$. Next, a continuous solution x_c is chosen from the uniform distribution $\mathcal{U}(0,1)$ and b is set to $b = Ax_c$. The problem is then turned into the format of (23), by setting $P = A^T A$ and $q = -A^T$.

6 Lovász Theta Function in Graph Theory

The Lovász theta function is an important concept in graph theory. Consider an undirected graph $G = (V, E)$ with vertex set $V = \{1, \dots, n\}$ and edge set $E \in \mathcal{S}^n$, i.e. $E_{ij} = 1$ if there exists an edge between the vertices i and j. A *stable set* (or *independent set*) is a subset of V such that the induced subgraph does not contain edges. The *stability number* $\alpha(G)$ of the graph is equal to the cardinality of the largest stable set. It is closely related to the *Shannon capacity* $\Theta(G)$ that models the amount of information that a noisy communication channel can carry if certain signal values can be confused with each other. Unfortunately, the determination of $\alpha(G)$ is a NP-hard problem and the computational complexity of $\Theta(G)$ remains unknown. The Lovász theta function $\vartheta(G)$ was introduced in [Lov79], can be computed in polynomial time, and represents an upper bound on both the stability number and the Shannon capacity:

$$\alpha(G) \leq \Theta(G) \leq \vartheta(G). \tag{27}$$

The value of $\vartheta(G)$ can be determined by solving a semidefinite program. It is equal to the optimal value p^* of the following problem:

$$\begin{aligned} \max_{X} \quad & \mathbf{Tr}(JX) \\ \text{s.t.} \quad & \mathbf{Tr}(X) = 1, \\ & X_{ij} = 0, \quad (i,j) \in E, \\ & X \succeq 0, \end{aligned} \tag{28}$$

with matrix variable $X \in R^{n \times n}$, edge set $E \in \mathcal{S}^n$, and matrix $J \in \mathbb{R}^{n \times n}$ of all ones.

For the test problems the number of vertices v for the undirected graph $G(V, E)$ is chosen from the interval $[10, 30]$. In a first step the edge matrix $E \in \mathcal{S}^n$ is chosen from the set of symmetric matrices with a density value in $[0.1, 0.9]$. In a second step all non-zero off-diagonal entries E_{ij} are set to one and all diagonal entries E_{ii} are set to 0.

References

[Lov79] László Lovász. "On the Shannon capacity of a graph". In: *IEEE Transactions on Information theory* 25.1 (1979), pp. 1–7.

[GA94] Pascal Gahinet and Pierre Apkarian. "A linear matrix inequality approach to Hinf control". In: *International journal of robust and nonlinear control* 4.4 (1994), pp. 421–448.

[VB96] Lieven Vandenberghe and Stephen Boyd. "Semidefinite programming". In: *SIAM review* 38.1 (1996), pp. 49–95.

[LKJ09] Daniel Lewandowski, Dorota Kurowicka, and Harry Joe. "Generating random correlation matrices based on vines and extended onion method". In: *Journal of multivariate analysis* 100.9 (2009), pp. 1989–2001.

[HM11] Didier Henrion and Jérôme Malick. "Projection methods for conic feasibility problems: applications to polynomial sum-of-squares decompositions". In: *Optimization Methods & Software* 26.1 (2011), pp. 23–46.

[PB17] Jaehyun Park and Stephen Boyd. "A semidefinite programming method for integer convex quadratic minimization". In: *Optimization Letters* (2017), pp. 1–20.