Smoothing techniques for solving semidefinite programs with many constraints

Michel Baes* Michael Bürgisser*

Technical report

Abstract

We use smoothing techniques to solve approximately mildly structured semidefinite programs with many constraints. As smoothing techniques require a specific problem format, we introduce an alternative problem formulation that fulfills the structural assumptions. The resulting algorithm has a complexity that depends linearly both on the number of constraints and on the inverse of the accuracy. Some numerical experiments show that smoothing techniques compares favorably with interior-point methods for very large-scale instances.

1 Introduction

A vast range of real-life optimization problems can be represented or approximated by semidefinite optimization problems, in such different fields as Control [BGFB94], Structural Design, or Statistics [BV04] to name a few. Of particular interest are semidefinite relaxations of hard combinatorial problems (see e.g. [GW95, NRT]). Semidefinite optimization, or semidefinite programming (SDP) can be seen as a generalization of linear programming (LP) [NRT]. LP constitutes indeed a particular set of SDP instances. Also, some algorithms for SDP can be understood as extensions of algorithms for LP, where, loosely speaking, the components of the unknown vector of variables have been replaced by the eigenvalues of the matrix variable [AS00].

SDP problems involving matrices of size of a few hundreds and with a few thousands constraints are solvable by interior-point methods up to a high accuracy. Denoting by n the size of the unknown matrix, by m the number of constraints, and by $\varepsilon > 0$ the desired absolute accuracy of the solution, interior-point methods have a theoretical worst-case running time of

$$\mathcal{O}\left(\left(mn^3 + m^2n^2 + m^3 + mn^2\right) n^{0.5} \ln(nC/\varepsilon)\right),$$

*M. Baes and M. Bürgisser are with the Institute for Operations Research, ETH, Rämistrasse 101, CH-8092 Zürich, Switzerland. E-Mail: {baes, buergisser}@ifor.math.ethz.ch.

where C is some constant depending on the starting point. In practice, interior-point methods are much more efficient, and, as a rule of thumb, the factor $n^{0.5}\ln(nC/\varepsilon)$ can usually be replaced by an absolute constant like 50 when $\varepsilon \approx 10^{-8}$ [BV96]. The complexity can be much more favorable in the case of a specially structured problem, e.g. when the instance data are sparse. Whereas the influence of the desired accuracy is modest, complexity is increasing rapidly as the size of the problem increases. As confirmed by extensive testings, interior-point methods are usually too costly in practice for dealing with very large scale problems.

A natural question arises: suppose that we accept to face a moderate complexity increase with respect to the accuracy; do algorithms with lower running time in m and n exist?

Arora and Kale [AK07] introduced an algorithm that has indeed a better complexity with respect to the problem size for some important classes of SDP problems. Their scheme performs a binary search over the objective function values. At each iteration, they have to establish whether the problem admits a feasible point that yields an objective value larger than the current guess. This operation is performed by a reasonably cheap procedure, which exhibits such a feasible point if it exists, or returns a dual feasible point that proves that the current guess is too high. It turns out that the cheap procedure they use is actually a particular instance of Nesterov's primal-dual subgradient method [Nes09].

Arora and Kale's cheap procedure requires at every iteration of their binary search method about $\mathcal{O}\left(\ln(n)/\varepsilon^2\right)$ iterations, each of which requires a matrix exponentiation. As about $\mathcal{O}(\ln(1/\varepsilon))$ binary search steps are required, the overall dependence of Arora and Kale's algorithm on ε is about $\mathcal{O}(\ln(1/\varepsilon)/\varepsilon^2)$, to be compared with $\mathcal{O}(\ln(1/\varepsilon))$ guaranteed by interior-point methods. Does there exist a method which benefits, as Arora and Kale's scheme, of a more favorable complexity with respect to the problem size than interior-point methods, but with a weaker influence of the desired accuracy? In this paper, we present an algorithm that can be applied for a fairly large class of SDPs and has a running time that depends linearly on the inverse of the relative accuracy while maintaining a reasonably modest iteration cost.

Nesterov [Nes05] developed a powerful machinery to solve some non-differentiable convex problems with a specific structure of non-differentiability. In a first step, an appropriate smooth approximation of the objective function is formed. Then, an optimal gradient method is applied to the smoothed version of the problem. In total, the algorithm requests $\mathcal{O}\left(1/\varepsilon\right)$ iterations of the optimal gradient scheme.

Chudak and Eleutério [CE05] implemented and tested this algorithm on large LP problems. Provided that the desired accuracy remains moderate, the resulting method outperforms CPLEX for problems with nine millions variables and millions of constraints. For a detailed description of their numerical results, we refer to [Ele09]. We can also mention the work of Peña [Peñ08], who uses Nesterov's smoothing to solve the Poker Game problem.

Since LP problems are special instances of SDP problems, we follow the

approach of Chudak and Eleutério [CE05, Ele09] and extend it to some (mildly) structured SDP problems. We reformulate such an SDP problem into a format that reveals its non-differentiability and renders the construction of its smooth approximation relatively easy. This smooth approximation is minimized using an optimal first-order method as in [Nes05]. The resulting algorithm has at the same time a better worst-case complexity with respect to the desired accuracy ε than Arora and Kale's method, and still outperforms interior-point methods regarding the dependence of the complexity on the problem size.

Iyengar et al. [IPS05, IPS09] have developed a similar procedure to solve some semidefinite packing problems, applying Nesterov's scheme to a different model than ours. Their construction is based on the Lagrangian relaxation of the initial problem, which, in their first paper [IPS05], originates from semidefinite relaxations of MaxCut and Graph Coloring. In their second paper [IPS09], they generalize their considerations to a broader class of problems.

This paper is structured as follows. We briefly discuss smoothing techniques in Section 2. We state the problem of interest in Section 3. Section 4 contains our reformulation of the original problem into a form more adapted to smoothing techniques. We apply smoothing techniques on a modified version of the generic model in Section 5 and discuss some of our implementation choices. We present some numerical results in Section 6.

2 Smoothing techniques

In this section, we review some of Nesterov's results on smoothing techniques [Nes05]. We assume that E_1 is a finite-dimensional real vector space, which is endowed with a norm denoted by $\|\cdot\|_{E_1}$. We write E_1^* for the dual space of E_1. In view of the celebrated Lax Theorem, we identify $\bar{w} \in E_1^*$ with the element $w \in E_1$ such that

$$\bar{w}(x) = \langle w, x \rangle_{E_1}$$

for any $x \in E_1$. We equip the dual space E_1^* with the norm

$$\|\bar{w}\|_{E_1}^* := \max_{x \in E_1} \{\langle w, x \rangle_{E_1} : \|x\|_{E_1} = 1\},$$

where $\bar{w} \in E_1^*$.

Let $Q_1 \subset E_1$ be a closed convex subset of E_1 and $f : Q_1 \to \mathbb{R}$ a convex function. We consider the problem of minimizing f over Q_1, possibly adding extra assumptions on f.

2.1 Optimal first-order method

In this subsection, we assume that the function f is differentiable on Q_1 and has a Lipschitz continuous gradient:

$$\|\nabla f(x) - \nabla f(y)\|_{E_1}^* \leq L \|x - y\|_{E_1}$$

for any $x, y \in Q_1$, where $L > 0$ denotes the Lipschitz constant with respect to the norm $\|\cdot\|_{E_1}$.

Let $d_1 : Q_1 \to \mathbb{R}$ be a *prox-function* for the set Q_1, i.e. a nonnegative strongly convex function with minimizer x_0 that belongs to the relative interior of Q_1. As d_1 is strongly convex, there exists a strong convexity parameter $\sigma_1 > 0$ for which:

$$\beta d_1(x) + (1-\beta)d_1(y) \geq d_1(\beta x + (1-\beta)y) + \frac{\sigma_1}{2}\beta(1-\beta)\|x-y\|_{E_1}^2$$

for any $x, y \in Q_1$ and $\beta \in [0,1]$. We assume that d_1 is differentiable on Q_1. Then, the strong convexity condition can be rewritten as

$$d_1(y) \geq d_1(x) + \langle \nabla d_1(x), y-x\rangle_{E_1} + \frac{1}{2}\sigma_1\|y-x\|_{E_1}^2$$

for any $x, y \in Q_1$. Without loss of generality, we may assume that the minimal value of d_1 over Q_1 is 0, i.e. $d_1(x_0) = 0$. The *diameter* of the set Q_1 is defined as:

$$D_1 := \max_{x \in Q_1} d_1(x).$$

We write

$$\xi(z,x) := d_1(x) - d_1(z) - \langle \nabla d_1(z), x-z\rangle_{E_1}$$

for the *Bregman distance* induced by d_1 between $x \in Q_1$ and $z \in Q_1$. Let

$$V_{Q_1}(z,g) := \arg\min_{x \in Q_1}\{\langle g, x-z\rangle_{E_1} + \xi(z,x)\},$$

where $z \in Q_1$ and $g \in E_1$. We assume that we can compute exactly and relatively cheaply the point $V_{Q_1}(z,g)$ for every z and g. Algorithm 2.1 minimizes the function f over Q. The following theorem, proved by Nesterov in [Nes05], shows how fast this algorithm converges to the exact minimum.

Theorem 2.1 *Assume that the sequences $(x_k)_{k=0}^{\infty}$, $(y_k)_{k=0}^{\infty}$, and $(z_k)_{k=0}^{\infty}$ are generated when applying Algorithm 2.1 to*

$$\min_{x \in Q_1} f(x). \tag{2.1}$$

Then,

$$f(y_k) - f(x^*) \leq \frac{4LD_1}{\sigma_1(k+1)(k+2)}$$

for any $k \geq 0$, where x^* is an optimal solution to Problem (2.1).

2.2 Smooth approximation of the objective function

In this subsection, we do not assume any more that the convex objective function is differentiable. Instead, we suppose that $f : Q_1 \to \mathbb{R}$ features a very specific nondifferentiability structure. This structure involves a compact convex set

Algorithm 2.1 Fast gradient scheme [Nes05]

1: Compute

$$y_0 := \arg\min_{x \in Q_1} \left\{ \frac{L}{\sigma_1} d_1(x) + \frac{1}{2} \left(f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle_{E_1} \right) \right\}.$$

2: **for** $k \geq 0$ **do**

3: Find

$$z_k := \arg\min_{x \in Q_1} \left\{ \frac{L}{\sigma_1} d_1(x) + \sum_{i=0}^{k} \frac{i+1}{2} \left(f(x_i) + \langle \nabla f(x_i), x - x_i \rangle_{E_1} \right) \right\}.$$

4: Set $\tau_k := \frac{2}{k+3}$ and $x_{k+1} := \tau_k z_k + (1 - \tau_k) y_k$.

5: Compute

$$\hat{x}_{k+1} := V_{Q_1}(z_k, \frac{\sigma_1 \tau_k}{L} \nabla f(x_{k+1})).$$

6: Set $y_{k+1} := \tau_k \hat{x}_{k+1} + (1 - \tau_k) y_k$.

7: **end for**

$Q_2 \subseteq E_2$, where E_2 is an appropriate Euclidean space of appropriate dimension endowed with a norm $\|\cdot\|_{E_2}$. We also need a linear operator $\bar{\mathcal{A}} : E_1 \to E_2^*$, where E_2^* is the dual of E_2. We identify $\bar{\mathcal{A}}$ with the mapping $\mathcal{A} : E_1 \to E_2$ that satisfies $\bar{\mathcal{A}}(x)(u) = \langle \mathcal{A}(x), u \rangle_{E_2}$ for any $x \in E_1$ and any $u \in E_2$. The adjoint operator $\bar{\mathcal{A}}^* : E_2 \to E_1^*$ of $\bar{\mathcal{A}}$ is defined as $\bar{\mathcal{A}}^*(u)(x) := \bar{\mathcal{A}}(x)(u)$ for any $x \in E_1$ and any $u \in E_2$. The norm of the linear operator $\bar{\mathcal{A}}$ is given by

$$\|\bar{\mathcal{A}}\|_{1,2} := \max_{x \in E_1,\ u \in E_2} \left\{ \langle \mathcal{A}(x), u \rangle_{E_2} : \|x\|_{E_1} = 1, \|u\|_{E_2} = 1 \right\}.$$

The objective function f is assumed to have the form:

$$f(x) := \hat{f}(x) + \max_{u \in Q_2} \left\{ \langle \mathcal{A}(x), u \rangle_{E_2} - \hat{\phi}(u) \right\}, \tag{2.2}$$

where the functions $\hat{f}$ and $\hat{\phi}$ are differentiable and convex on Q_1 and on Q_2, respectively. Thus, our problem can be written as

$$\min_{x \in Q_1} \left\{ f(x) = \hat{f}(x) + \max_{u \in Q_2} \left\{ \langle \mathcal{A}(x), u \rangle_{E_2} - \hat{\phi}(u) \right\} \right\}. \tag{2.3}$$

We can write (2.3) in the dual form:

$$\max_{u \in Q_2} \left\{ \phi(u) := -\hat{\phi}(u) + \min_{x \in Q_1} \left\{ \langle \mathcal{A}(x), u \rangle_{E_2} + \hat{f}(x) \right\} \right\}.$$

Now, we can form a smooth approximation of f to which we can apply the fast gradient scheme in Algorithm 2.1. Let d_2 be a prox-function of the set Q_2, with σ_2 as strong convexity parameter with respect to $\|\cdot\|_{E_2}$. We denote by u_0

its *prox-center*, that is, its minimizer on Q_2, and we assume that $d_2(u_0) = 0$. We define the *diameter* of Q_2 as:

$$D_2 := \max_{u \in Q_2} d_2(u).$$

Consider the following function:

$$f_\mu(x) := \max_{u \in Q_2} \left\{ \langle \mathcal{A}(x), u \rangle_{E_2} - \hat{\phi}(u) - \mu d_2(u) \right\} \tag{2.4}$$

defined on Q_1 and where $\mu > 0$ is a positive smoothness parameter. The function f_μ is a uniform approximation of

$$\max_{u \in Q_2} \left\{ \langle \mathcal{A}(x), u \rangle_{E_2} - \hat{\phi}(u) \right\},$$

because

$$f_\mu(x) \leq \max_{u \in Q_2} \left\{ \langle \mathcal{A}(x), u \rangle_{E_2} - \hat{\phi}(u) \right\} \leq f_\mu(x) + \mu D_2$$

for any $x \in E_1$. We write $u(x)$ for the optimal solution to Problem (2.4). Note that this solution is unique since the prox-function d_2 is strongly convex.

Theorem 2.2 [Nes05] *The function f_μ is well-defined, continuously differentiable, and convex on E_1. The gradient of f_μ takes the form*

$$\nabla f_\mu(x) = \bar{\mathcal{A}}^*(u(x)),$$

and is Lipschitz continuous with the constant

$$L_\mu := \frac{1}{\mu \sigma_2} \|\bar{\mathcal{A}}\|_{1,2}^2.$$

Thus, we may run Algorithm 2.1 on the following problem:

$$\min_{x \in Q_1} f_\mu(x). \tag{2.5}$$

Theorem 2.3 [Nes05] *Let $N \geq 0$ and*

$$\mu = \mu(N) := \frac{2\|\bar{\mathcal{A}}\|_{1,2}}{N+1} \sqrt{\frac{D_1}{\sigma_1 \sigma_2 D_2}}.$$

We denote by $(x_k)_{k=0}^N$, $(y_k)_{k=0}^N$, and $(z_k)_{k=0}^N$ the sequences generated by Algorithm 2.1 when applied to Problem (2.5). Then,

$$0 \leq f(\bar{x}) - \phi(\bar{u}) \leq \frac{4\|\bar{\mathcal{A}}\|_{1,2}}{N+1} \sqrt{\frac{D_1 D_2}{\sigma_1 \sigma_2}},$$

where $\bar{x} := y_N$ and

$$\bar{u} := \sum_{k=0}^{N} \frac{2(k+1)}{(N+1)(N+2)} u(x_k).$$

For the n-dimensional standard simplex

$$\Delta_n := \left\{ x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1,\ x \geq 0 \right\},$$

a particularly judicious choice of prox-function is

$$d : \Delta_n \to \mathbb{R} : x \mapsto \ln(n) + \sum_{i=1}^n x_i \ln(x_i).$$

Indeed, the function d has a strong convexity parameter equal to 1 with respect to the best possible norm, that is, the 1-norm $\|\cdot\|_1$ (see e.g. [Nes05] for a proof).

3 Semidefinite programming

Let $\mathcal{M}_n$ be the space of real matrices of dimension $n \times n$. The function

$$\mathrm{Tr} : \mathcal{M}_n \to \mathbb{R}$$

represents the trace of its argument. The standard Frobenius scalar product is denoted by $\langle \cdot, \cdot \rangle_F$. We write $\mathcal{S}_n$ for the $n(n+1)/2$-dimensional vector subspace of real symmetric matrices in $\mathcal{M}_n$.

Let $X \in \mathcal{S}_n$. We write $X \succeq 0$ [resp. $X \succ 0$] if X is positive semidefinite [resp. positive definite], i.e. if $v^T X v \geq 0$ for every $v \in \mathbb{R}^n$ [resp. $v^T X v > 0$ for each nonzero $v \in \mathbb{R}^n$].

3.1 A semidefinite optimization problem

A wide class of semidefinite optimization problems can be written in the following form:

$$\begin{array}{ll} \max\limits_X & \langle C, X \rangle_F \\ \text{s. t.} & \langle A_j, X \rangle_F \leq 1 \qquad \text{for any } j = 1, \ldots, m \\ & X \succeq 0, \end{array} \tag{3.1}$$

where $A_1, \ldots, A_m, C \in \mathcal{S}_n$. The dual of (3.1) takes the following form:

$$\begin{array}{ll} \min\limits_y & \sum_{j=1}^m y_j \\ \text{s. t.} & \sum_{j=1}^m y_j A_j \succeq C \\ & y \geq 0. \end{array}$$

For more details about duality theory we refer to [BV04]. In this paper, we make the following assumptions on the above general semidefinite optimization problem.

Assumption 1: The matrix C is positive definite.

Assumption 2: There exists $\bar{y} \in \mathbb{R}^m$ with $\bar{y} \geq 0$ and $\sum_{j=1}^m \bar{y}_j A_j \succeq C$.

These two assumptions ensure that our optimization problem has an optimal solution, and that the optimal value is positive. Indeed, let $\alpha > 0$ such that $\max_{j=1,\ldots,m} \mathrm{Tr}(A_j) < \alpha$. Denoting by $\mathbb{I}_n$ the identity matrix in $\mathcal{M}_n$, the matrix $\frac{1}{\alpha}\mathbb{I}_n$ is strictly feasible for (3.1). Therefore, we have by the first assumption:

$$\langle C, \frac{1}{\alpha}\mathbb{I}_n \rangle_F = \frac{1}{\alpha}\mathrm{Tr}(C) > 0.$$

By the second assumption, there exists a dual feasible solution $\bar{y}$ with finite objective function value $\sum_{j=1}^m \bar{y}_j < \infty$. We obtain by weak duality that the primal problem (3.1) is bounded. Thus, there exists an optimal solution $\hat{X}$ to Problem (3.1) with $\langle C, \hat{X} \rangle_F > 0$.

3.2 Space of symmetric matrices: norms and dual space

We denote by $\lambda(X)$ the vector of eigenvalues of the matrix $X \in \mathcal{S}_n$. Conventionally, we assume that the components of $\lambda(X)$ are ordered decreasingly.

Given a vector $\lambda \in \mathbb{R}^n$, the diagonal matrix $D(\lambda)$ represents the matrix whose diagonal is λ. Every $X \in \mathcal{S}_n$ admits an eigendecomposition:

$$X = Q(X)D(\lambda(X))Q(X)^T = \sum_{i=1}^n \lambda_i(X) q_i(X) q_i(X)^T,$$

where $Q(X) := (q_1(X), \ldots, q_n(X))$ is a (not necessarily unique) orthogonal matrix of dimension $n \times n$, and $q_1(X), \ldots, q_n(X)$ are unitary eigenvectors corresponding respectively to the eigenvalues $\lambda_1(X), \ldots, \lambda_n(X)$. An eigendecomposition of a generic symmetric matrix can be computed in $\mathcal{O}(n^3)$ elementary operations [HJ96].

Some matrix norms are defined through eigenvalues. For instance, the *induced p-norms* are:

$$\|X\|_p := \sqrt[p]{\sum_{i=1}^n |\lambda_i(X)|^p},$$

where $1 \leq p < \infty$ and $X \in \mathcal{S}_n$, and

$$\|X\|_\infty := \max_{i=1,\ldots,n} |\lambda_i(X)|,$$

which is the limit of $\|X\|_p$ when p goes to infinity.

We endow the space $\mathcal{S}_n$ with a norm $\|\cdot\|$. The set of all real-valued linear forms on $\mathcal{S}_n$, that is, the dual space of $\mathcal{S}_n$, is denoted as $\mathcal{S}_n^*$. If we interpret the elements of $\mathcal{S}_n$ as column vectors, the elements of $\mathcal{S}_n^*$ can be seen as row vectors of length $n(n+1)/2$, and the dual space of $\mathcal{S}_n$ is isomorphic to $\mathcal{S}_n$ itself. Let $\bar{W}$ be an element of the dual space $\mathcal{S}_n^*$. As done earlier in this paper, we identify $\bar{W}$ with the unique element $W \in \mathcal{S}_n$ for which $\bar{W}(X) = \langle W, X \rangle_F$ for all $X \in \mathcal{S}_n$. The norm of the dual space is defined in the following way:

$$\|\bar{W}\|^* \quad := \quad \max_{X \in \mathcal{S}_n} \left\{ \bar{W}(X) : \|X\| = 1 \right\}$$

$$= \max_{X \in \mathcal{S}_n} \{\langle W, X\rangle_F : \|X\| = 1\},$$

where $\bar{W} \in \mathcal{S}_n^*$.

4 The generic model

In this section, we transform Problem (3.1) into a format to which the smoothing techniques presented in Section 2 can be applied.

4.1 Problem transformation

In view of our assumptions, the quantity $\max_{j=1,\ldots,m}\langle A_j, X\rangle_F$ is positive for every nonzero $X \succeq 0$. Assume on the contrary that this maximum is nonpositive for a $X \succeq 0$ different from 0. On the one hand, the objective $\langle C, X\rangle_F$ is nonpositive in view of Assumption 2. On the other hand, $\langle C, X\rangle_F \geq 0$ because $C, X \succeq 0$. Thus $\langle C, X\rangle_F$ must be null. As $C \succ 0$ and $X \neq 0$, we get a contradiction.

We obtain:

$$\frac{\langle C, X\rangle_F}{\max_{j=1,\ldots,m}\langle A_j, X\rangle_F} \in [0, \infty)$$

for every nonzero $X \succeq 0$.

The following simple lemma constitutes our first step in the reformulation of our problem.

Lemma 4.1 *In view of our assumptions, we have:*

$$\max_{X \succeq 0} \left\{ \langle C, X\rangle_F : \max_{j=1,\ldots,m} \langle A_j, X\rangle_F \leq 1 \right\} = \left(\min_{Y \in Q'} \max_{j=1,\ldots,m} \langle A_j, Y\rangle_F \right)^{-1}, \quad (4.1)$$

where $Q' := \{Y \succeq 0 : \langle C, Y\rangle_F = 1\}$.

Proof

Let us first prove the following equality:

$$\alpha := \max_{X \succeq 0} \left\{ \langle C, X\rangle_F : \max_{j=1,\ldots,m} \langle A_j, X\rangle_F \leq 1 \right\} = \sup_{X \succeq 0} \frac{\langle C, X\rangle_F}{\max_{j=1,\ldots,m}\langle A_j, X\rangle_F} =: \beta.$$

Obviously, $\alpha \leq \beta$. Indeed, if $\hat{X}$ is an optimal solution of the left-hand side problem, then:

$$\max_{j=1,\ldots,m} \langle A_j, \hat{X}\rangle_F = 1$$

due to the linearity of the objective function. Hence:

$$\langle C, \hat{X}\rangle_F = \frac{\langle C, \hat{X}\rangle_F}{\max_{j=1,\ldots,m}\langle A_j, \hat{X}\rangle_F} \leq \beta.$$

On the other hand, let $\epsilon > 0$ and $\bar{X}$ be such that

$$\beta - \epsilon \leq \frac{\langle C, \bar{X} \rangle_F}{\max_{j=1,\ldots,m} \langle A_j, \bar{X} \rangle_F}.$$

Let $X_\epsilon := \bar{X} / \max_{j=1,\ldots,m} \langle A_j, \bar{X} \rangle_F$. Evidently, X_ϵ is feasible for the left-hand side problem of (4.1), and $\beta - \epsilon \leq \langle C, X_\epsilon \rangle_F \leq \alpha$. Therefore $\beta \leq \alpha$ and $\alpha = \beta$.

Observe that Y belongs to Q' if and only if there exists $X \succeq 0$ such that:

$$Y = \frac{X}{\langle C, X \rangle_F}.$$

Also, $\beta > 0$ implies that we can write

$$\sup_{X \succeq 0} \left\{ \frac{\langle C, X \rangle_F}{\max_{j=1,\ldots,m} \langle A_j, X \rangle_F} \right\} = \max_{Y \in Q'} \left(\max_{j=1,\ldots,m} \langle A_j, Y \rangle_F \right)^{-1},$$

and this last problem is equivalent to the right-hand side of (4.1). ■

We have converted our original problem into the following:

$$\min_{Y \in Q'} \max_{j=1,\ldots,m} \langle A_j, Y \rangle_F = \min_{Y \in Q'} \max_{u \in \Delta_m} \sum_{j=1}^{m} u_j \langle A_j, Y \rangle_F.$$

Recall that Δ_m is the m-dimensional simplex.

Note that the objective function is not differentiable anymore. Moreover, *the non-differentiability enters in the problem through a maximization over a compact convex set.* This is exactly the format needed by the smoothing techniques by Nesterov.

We can simplify our problem further by performing a Cholesky decomposition of the positive definite matrix C. Let $C = LL^T$, where $L \in \mathcal{M}_n$ is lower triangular. Since C is positive definite, L is invertible, as well. About $\mathcal{O}(n^3)$ elementary operations are needed to compute such a matrix L.

Let $Y \in Q'$, and let $\bar{Y} := L^T Y L \succeq 0$. We have, by invertibility of L:

$$\langle C, Y \rangle_F = \langle LL^T, L^{-T} \bar{Y} L^{-1} \rangle_F = \mathrm{Tr}(\bar{Y}),$$

and

$$\langle A_j, Y \rangle_F = \langle A_j, L^{-T} \bar{Y} L^{-1} \rangle_F = \langle L^{-1} A_j L^{-T}, \bar{Y} \rangle_F$$

for any $j = 1, \ldots, m$. Writing as

$$\Delta_n^M := \{ X \succeq 0 : \ \mathrm{Tr}(X) = 1 \},$$

for the standard simplex in matrix form, we have proved the following proposition.

Proposition 4.1 *Let $B_j := L^{-1} A_j L^{-T}$. We have:*

$$\max_{X \succeq 0} \left\{ \langle C, X \rangle_F : \max_{j=1,\ldots,m} \langle A_j, X \rangle_F \leq 1 \right\} = \left(\min_{X \in \Delta_n^M} \max_{j=1,\ldots,m} \langle B_j, X \rangle_F \right)^{-1}.$$

Therefore, the problem consists in solving:

$$\min_{X\in\Delta_n^M}\max_{j=1,\ldots,m}\langle B_j,X\rangle_F=\min_{X\in\Delta_n^M}\max_{u\in\Delta_m}\sum_{j=1}^m u_j\langle B_j,X\rangle_F. \tag{4.2}$$

In the remaining of this paper, we refer to the previous problem as the *generic model*.

Note that the last transformation we perform can affect the possible structure of the original problem. For instance, the matrices B_j might be dense even though the matrices A_j can be sparse.

4.2 The dual of the generic problem

The generic model can take two equivalent forms, namely the *primal*, which is the original one:

$$\min_{X\in\Delta_n^M}\max_{u\in\Delta_m}\langle\sum_{j=1}^m u_jB_j,X\rangle_F,$$

and the *dual* one:

$$-\min_{u\in\Delta_m}\max_{X\in\Delta_n^M}\langle-\sum_{j=1}^m u_jB_j,X\rangle_F. \tag{4.3}$$

The equivalence of these two problems comes from the standard Minimax Theorem in Convex Analysis, and is due to the fact that the simplex Δ_m is compact (see Corollary 37.3.2 in [Roc70] for a proof of the Minimax Theorem). We leave the discussion on which problem is more advantageous to solve using smoothing techniques for Section 5.

Interestingly, the dual is also equivalent to an eigenvalue optimization problem. The following proposition is an immediate consequence of the well-known Fan's Inequalities [Fan49].

Proposition 4.2 *Let $G\in\mathcal{S}_n$. Then $\lambda_1(G)=\max_{X\in\Delta_n^M}\langle G,X\rangle_F$.*

In view of this lemma, the dual can be rewritten as the following eigenvalue optimization problem:

$$-\min_{u\in\Delta_m}\lambda_1\left(-\sum_{j=1}^m u_jB_j\right).$$

4.3 Constructing a solution with guaranteed relative accuracy

For the sake of clarity, let us briefly define what we understand by relative accuracy. Let $g:U\to\mathbb{R}$ be a function, and let us assume that the supremum g^* of g over U is bounded. Given an accuracy $\varepsilon>0$, we call an element $u\in U$ an *ε-solution* if

$$g^*-g(u)\leq\varepsilon.$$

Suppose that $g^* > 0$. We say that $u \in U$ is an *approximate solution* to $\max_{u\in U} g(u)$ *with relative accuracy* $\delta > 0$, if

$$g^* - g(u) \leq \delta g^*.$$

Proposition 4.3 *Let $\varepsilon > 0$ and X^ε be an ε-solution to the generic model (4.2). The matrix:*

$$X_\delta := \frac{L^{-T} X^\varepsilon L^{-1}}{\max_{j=1,\ldots,m} \langle A_j, L^{-T} X^\varepsilon L^{-1} \rangle_F} = \frac{L^{-T} X^\varepsilon L^{-1}}{\max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F} \tag{4.4}$$

is well-defined and represents an approximate solution to Problem (3.1) with relative accuracy

$$\delta = \frac{\varepsilon}{\max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F} > 0.$$

Moreover,

$$\langle C, X_\delta \rangle_F \max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F = 1. \tag{4.5}$$

Proof
The matrix X_δ is well-defined, because the denominator $\max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F$ is positive. Indeed, if this denominator was not positive, we would have:

$$\min_{X\in\Delta_n^M} \max_{j=1,\ldots,m} \langle B_j, X \rangle_F \leq \max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F \leq 0.$$

However, according to Proposition 4.1, we have:

$$\left(\min_{X\in\Delta_n^M} \max_{j=1,\ldots,m} \langle B_j, X \rangle_F \right)^{-1} = \max_{X\succeq 0} \left\{ \langle C, X \rangle_F : \max_{j=1,\ldots,m} \langle A_j, X \rangle_F \leq 1 \right\},$$

implying that the optimum of the right-hand side problem is not positive. But this contradicts Assumption 2.

As X^ε is feasible for the generic model (4.2), the matrix X_δ is positive semidefinite. Moreover, $\max_{j=1,\ldots,m} \langle A_j, X_\delta \rangle_F = 1$, and therefore X_δ is a feasible point of Problem (3.1). We can easily see that:

$$\begin{aligned} \langle C, X_\delta \rangle_F \max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F &= \frac{\langle C, L^{-T} X^\varepsilon L^{-1} \rangle_F}{\max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F} \max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F \\ &= \mathrm{Tr}(X^\varepsilon) = 1. \end{aligned}$$

For the sake of notational simplicity, let us define the function:

$$\phi : \mathcal{S}_n \to \mathbb{R}, \quad X \mapsto \phi(X) := \max_{j=1,\ldots,m} \langle B_j, X \rangle_F.$$

That is in fact the objective function of the generic model, and we write ϕ^* for its minimal value on Δ_n^M. Since X^ε is an ε-solution to (4.2), it holds that

$$\phi(X^\varepsilon) - \phi^* \leq \varepsilon.$$

Thus, by Theorem 4.1 and (4.5), we obtain:

$$\frac{1}{\phi^*} - \langle C, X_\delta \rangle_F = \frac{1}{\phi^*} - \frac{1}{\phi(X^\varepsilon)} \leq \frac{\varepsilon}{\phi^* \phi(X^\varepsilon)},$$

and we can conclude. ■

Assume that we have a procedure to compute solutions of the generic model (4.2) with a fixed absolute accuracy $\varepsilon > 0$. The previous proposition shows how to construct an approximate solution to the original problem (3.1), and to evaluate its relative accuracy δ.

Suppose now that we choose the relative accuracy $\delta > 0$ we are interested in, and that we have a procedure to construct approximations of the solution to the generic model (4.2) with a specified absolute accuracy. Let $\epsilon := \delta / \sum_{j=1}^m \bar{y}_j$, for the $\bar{y} \geq 0$ delivered by Assumption 2, and let X^ε be the matrix returned by the procedure for the absolute accuracy ε. We construct the matrix X_δ as in (4.4). This matrix is an approximate solution of the original problem with relative accuracy δ. Indeed, since X_δ is feasible, we can write:

$$\langle C, X_\delta \rangle_F \leq \sum_{j=1}^m \bar{y}_j \langle A_j, X_\delta \rangle_F \leq \sum_{j=1}^m \bar{y}_j.$$

Therefore,

$$\varepsilon = \frac{\delta}{\sum_{j=1}^m \bar{y}_j} \leq \frac{\delta}{\langle C, X_\delta \rangle_F} = \delta \max_{j=1,\ldots,m} \langle B_j, X^\varepsilon \rangle_F.$$

The last equality comes from (4.5). Proposition 4.3 entails that X_δ is a solution to (3.1) with a relative accuracy of at least δ.

5 Smoothing and solving the generic model

We have shown in Subsection 4.2 that the generic model can take two different forms, the primal and the dual. After specifying a smoothing procedure for either problems, we study in this section some computational aspects of the smoothing algorithm applied to them.

5.1 Smooth approximation of the primal and of the dual

As mentioned in Section 2, the objective function of the primal can be smoothed as:

$$F_\mu(X) := \max_{u \in \Delta_m} \sum_{j=1}^m u_j \langle B_j, X \rangle_F - \mu \left(\sum_{j=1}^m u_j \ln(u_j) + \ln(m) \right). \tag{5.1}$$

We can easily derive a closed-form formula for the value of this function, and for its gradient (see e.g. [Nes05]):

$$F_\mu(X) = \mu \ln \left(\sum_{j=1}^m \exp \left(\frac{\langle B_j, X \rangle_F}{\mu} \right) \right) - \mu \ln(m),$$

$$\nabla F_\mu(X) = \sum_{j=1}^{m} \frac{B_j \exp\left(\langle B_j, X\rangle_F/\mu\right)}{\sum_{k=1}^{m} \exp\left(\langle B_k, X\rangle_F/\mu\right)}.$$

The computation of the gradient requires $\mathcal{O}(mn^2)$ operations.

Now, the smoothing algorithm requires a prox-function for the feasible set Δ_n^M. According to Nesterov [Nes07], the most appropriate one is the matrix entropy function:

$$D : \Delta_n^M \to \mathbb{R}, \quad X \mapsto D(X) := \ln(n) + \sum_{i=1}^{n} \lambda_i(X) \ln(\lambda_i(X)).$$

This function is strongly convex with constant 1 respect to the induced 1-norm, and the diameter of Δ_n^M equals $\ln(n)$. Its prox-center is $\mathbb{I}_n/n$.

We need to solve at every iteration of the smoothing algorithm two optimization problems of the form

$$\min_{X \in \Delta_n^M} \langle G, X\rangle_F - D(X),$$

where G is a symmetric matrix. The solution of this problem can also be computed explicitly, as sketched in the Appendix. As we need to perform a matrix exponentiation, the corresponding cost goes up to $\mathcal{O}(n^3)$.

We turn now our attention to the dual strategy. The computation cost of the gradient of the smoothed function, namely of:

$$f_\mu(u) := \max_{X \in \Delta_n^M} \langle -\sum_{j=1}^{m} u_j B_j, X\rangle_F - \mu D(X),$$

is significantly higher. By Theorem 2.2,

$$[\nabla f_\mu(u)]_j = -\langle B_j, X(u)\rangle_F$$

where $X(u)$ is the maximizer of the above maximization problem. In contrast, the two optimization subproblems require only $\mathcal{O}(m)$ elementary iterations to be solved. This leaves a clear advantage for the dual formulation of our generic problem.

5.2 Iteration number and complexity of the fast gradient scheme

Let us now evaluate the number of iterations needed by the smoothing algorithm on the dual problem. In view of Theorem 2.3, we need to estimate:

$$\|\bar{\mathcal{A}}\|_{1,2} := \max_{u \in \mathbb{R}^m, X \in \mathcal{S}_n} \left\{\langle \mathcal{A}(u), X\rangle_F : \|u\|_1 = 1, \|X\|_1 = 1\right\},$$

where:

$$\mathcal{A} : \mathbb{R}^m \to \mathcal{S}_n, \quad u \mapsto \mathcal{A}(u) := \sum_{j=1}^{m} u_j B_j,$$

and the norms are respectively the 1-vector norm and the induced 1-norm. We have:

$$\begin{aligned}
\|\bar{\mathcal{A}}\|_{1,2} &= \max_{u\in\mathbb{R}^m, X\in\mathcal{S}_n}\left\{\langle -\sum_{j=1}^m u_j B_j, X\rangle_F : \|u\|_1 = 1, \|X\|_1 = 1\right\} \\
&= \max_{u\in\mathbb{R}^m}\left\{\lambda_1\left(-\sum_{j=1}^m u_j B_j\right) : \|u\|_1 = 1\right\} \\
&= \max_{j=1,\ldots,m} |\lambda_1(-B_j)| \\
&= \max_{j=1,\ldots,m} \|B_j\|_\infty.
\end{aligned}$$

We used the fact that the (induced) ∞-norm is the dual of the (induced) 1-norm, see in the Appendix for more details. We can therefore guarantee that the smoothed function has a gradient Lipschitz constant of:

$$L_\mu := \frac{1}{\mu}\max_{j=1,\ldots,m}\|B_j\|_\infty^2. \tag{5.2}$$

We choose a relative accuracy $\delta > 0$ and we set the smoothness parameter at:

$$\mu := \frac{\delta}{2\ln(n)\|\bar{y}\|_1},$$

where $\bar{y}$ is given by Assumption 2. Denote by N the number of iterations. Let $(x_k)_{k=0}^N$, $(y_k)_{k=0}^N$, and $(z_k)_{k=0}^N$ be the sequences generated by Algorithm 2.1. Furthermore, for $u \in \Delta_m$, we write $X(u)$ for the maximizer of the problem:

$$\max_{X\in\Delta_n^M}\left\{\langle -\sum_{j=1}^m u_j B_j, X\rangle_F - \mu D(X)\right\}.$$

Defining

$$\bar{u} := y_N \text{ and } \bar{X} := \sum_{k=0}^N \frac{2(k+1)}{(N+1)(N+2)} X(u_k), \tag{5.3}$$

we have the following result.

Theorem 5.1 *If*

$$N \geq \frac{4}{\delta}\left(\max_{j=1,\ldots,m}||B_j||_\infty \|\bar{y}\|_1\sqrt{\ln(m)\ln(n)}\right),$$

then

$$X_\delta := \frac{L^{-T}\bar{X}L^{-1}}{\max_{j=1,\ldots,m}\langle A_j, L^{-T}\bar{X}L^{-1}\rangle_F}$$

is an approximate solution to Problem (3.1) with relative accuracy δ.

Proof
Defining

$$N := \frac{4}{\delta}\left(\max_{j=1,\dots,m}\|B_j\|_\infty \|\bar{y}\|_1 \sqrt{\ln(m)\ln(n)}\right) - 1$$

and applying Theorem 2.3, we obtain:

$$\max_{X\in\Delta_n^M} \langle -\sum_{j=1}^m \bar{u}_j B_j, X\rangle_F - \min_{u\in\Delta_m} \langle -\sum_{j=1}^m u_j B_j, \bar{X}\rangle_F \leq \frac{\delta}{\|\bar{y}\|_1}.$$

Thus, we have:

$$\max_{j=1,\dots,m} \langle B_j, \bar{X}\rangle_F - \min_{X\in\Delta_n^M} \max_{j=1,\dots,m} \langle B_j, \bar{X}\rangle_F \leq \frac{\delta}{\|\bar{y}\|_1}.$$

It remains to use Theorem 4.3. ■

It is now easy to establish the full complexity result.

Theorem 5.2 *We can approximately solve Problem (3.1) with relative accuracy* $\delta > 0$ *in*

$$\mathcal{O}\left(\frac{n^3 + mn^2}{\delta} \max_{j=1,\dots,m}\|B_j\|_\infty \|\bar{y}\|_1 \sqrt{\ln(m)\ln(n)} + mn^3\right)$$

elementary operations.

Observe that we can replace $\|\bar{y}\|_1$ by any upper bound on the maximizer of the original problem.

5.3 A consideration on numerical stability

When evaluating the gradient of the smoothed objective function or the solution of subproblems, it might happen that we have to compute the ratio of two exceedingly small numbers. In order to avoid numerical problems, we proceed as suggested in [Nes05]. When we need to evaluate an expression of the form:

$$\frac{\exp(-g_j/\gamma)}{\sum_{k=1}^m \exp(-g_k/\gamma)},$$

we compute:

$$\frac{\exp((g_{\min} - g_j)/\gamma)}{\sum_{k=1}^m \exp((g_{\min} - g_k)/\gamma)},$$

where $g_{\min}$ is the minimal component of the vector g. One term of the sum in the denominator equals one, and the other are smaller.

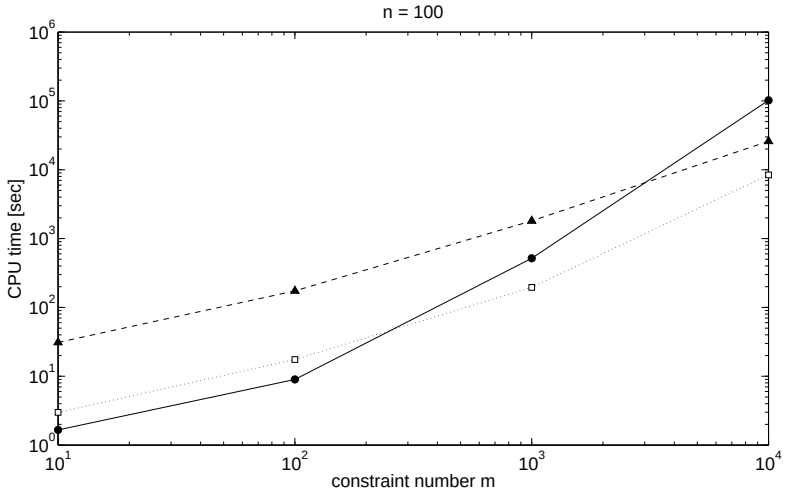

Figure 6.1: CPU time needed to solve randomly generated instances of Problem (3.1): The dots, triangles, and squares represent the CPU times needed by SeDuMi, smoothing techniques with Algorithm 2.1, and smoothing techniques with Algorithm 2.1 and with an L_μ that is adjusted by 1/256, respectively. The corresponding numerical values are shown in Tables A.1 and A.2.

6 Numerical results

In order to test our method, we solve randomly generated instances of Problem (3.1). In particular, the randomly generated matrices $A_1, \ldots, A_m$, and C are dense. We set $\delta := 0.01$ and apply smoothing techniques with Algorithm 2.1 to solve the instances of Problem (3.1). We check every 100 iterations (or every 10 iterations for small problems) whether:

$$\max_{j=1,\ldots,m} \langle B_j, \bar{X} \rangle_F - \min_{X \in \Delta_n^M} \langle \sum_{j=1}^{m} \bar{u}_j B_j, X \rangle_F \leq \delta \max_{j=1,\ldots,m} \langle B_j, \bar{X} \rangle_F, \tag{6.1}$$

where the current iterations $\bar{X}$ and $\bar{u}$ are defined in (5.3). If this inequality is satisfied, then X_δ defined in Theorem 5.1 is an approximate solution to (3.1) with relative accuracy δ. When running Algorithm 2.1 in Matlab, we call the built-in function `eig` whenever we need an eigendecomposition of a matrix.

For the computations, we use a computer with two AMD Opteron processors with a CPU of 2.2 GHz, and with 3.6 GB of RAM. For comparing smoothing techniques with interior-point methods, we use SeDuMi, a free software available on the internet. It implements a primal-dual interior point method.

The numerical results are presented in Figure 6.1 (and in Table A.1). We see that SeDuMi shows a better performance than Algorithm 2.1 with respect to the CPU time for problem instances involving matrices of dimension up to

100×100 and up to $1'000$ constraints. For problems with more constraints (and that involve matrices of dimension 100×100), Algorithm 2.1 outperforms SeDuMi, in spite of the fact that SeDuMi is written in C, while our version of Algorithm 2.1 is in Matlab.

The computation of the Lipschitz constant L_μ in (5.2) is done by applying Theorem 2.2. Having a closer look at the proof of Theorem 2.2, see [Nes05], we observe that L_μ is rather an upper bound on the Lipschitz constant of the gradient of the smooth objective function f_μ. The numerical results in Figure 6.1 (and in Table A.2) show that we may deal with an adjusted L_μ, and that the required CPU times are much more favorable. Of course, from a theoretical point of view, we are not guaranteed that the adjusted L_μ is an upper bound on the Lipschitz constant ∇f_μ.

Acknowledgments: We would like to thank Hans-Jakob Lüthi and Yurii Nesterov for the helpful discussions. This research is partially funded by the Swiss National Science Foundation (SNF), project no 111810, "Combinatorial algorithms for approximately solving classes of large scale linear programs and applications".

A Variational analysis of eigenvalues and gradient computation

In order to compute the gradient of the smoothed dual objective, we need the following result by Lewis. For every vector $\lambda \in \mathbb{R}^n$, we adopt the notation $\lambda_\downarrow$ for the vector containing the coefficients of λ ordered decreasingly.

Lemma A.1 ([Lew96]) *Let $G \in \mathcal{S}_n$ and $\lambda \in \Delta_n$. Then, the optimal value of the problem*

$$\min_{X \in \Delta_n^M : \lambda(X) = \lambda_\downarrow} \langle G, X \rangle_F$$

is attained at an $X \in \Delta_n^M$ that commutes with G, i.e. $XG = GX$.

We consider the problem of finding

$$X^* := \arg\min_{X \in \Delta_n^M} \left\{ \langle G, X \rangle_F + \gamma \sum_{i=1}^n \lambda_i(X) \ln(\lambda_i(X)) \right\}, \tag{A.1}$$

where $\gamma > 0$ is some constant and $G \in \mathcal{S}_n$. We may rewrite this problem under the form:

$$\min_{\lambda \in \Delta_n} \left\{ \min_{X \in \Delta_n^M : \lambda(X) = \lambda_\downarrow} \langle G, X \rangle_F + \gamma \sum_{i=1}^n \lambda_i \ln(\lambda_i) \right\},$$

since the order of the eigenvalues has no impact on the value of the second summand. Lemma A.1 tells us that an optimal solution X to the inner problem

$$\min_{X \in \Delta_n^M :\ \lambda(X) = \lambda\downarrow} \langle G, X \rangle_F$$

m	n	SeDuMi	Alg. 2.1	opt. ratio	N_{theory}	N_{used}
10	10	0.07 sec	0.61 sec	0.9915	$1'771$	$1'260$
10	100	1.66 sec	31.01 sec	0.9901	$4'559$	$1'100$
100	100	8.99 sec	173.54 sec	0.9924	$4'842$	$2'200$
$1'000$	100	517.47 sec	$1'807.13$ sec	0.9950	$5'497$	$3'000$
$10'000$	100	$101'994.61$ sec	$25'897.08$ sec	0.9964	$5'969$	$3'400$

Table A.1: *We solve randomly generated instances of Problem (3.1) by SeDuMi and by smoothing techniques with Algorithm 2.1. From left to right, the columns correspond to: the constraint number m, the matrix size n, the CPU time needed by SeDuMi, the CPU time needed by Algorithm 2.1 to solve a smoothed version of Problem (3.1), the optimality ratio that is the ratio between the objective function value computed by Algorithm 2.1 and the output of SeDuMi , the theoretical number of iterations required by Algorithm 2.1, and the number of iterations performed by Algorithm 2.1 until condition (6.1) is satisfied.*

commutes with G, or equivalently that G and X are simultaneously diagonalizable. Considering an eigendecomposition of G:

$$G = Q(G)D(\lambda(G))Q(G)^T,$$

where $D(\lambda(G))$ is a diagonal matrix with $\lambda(G)$ on its diagonal and $Q(G)$ an orthogonal matrix, we need to find the optimal solution λ^* to the problem:

$$\min_{\lambda \in \Delta_n} \langle \lambda(G), \lambda \rangle + \gamma \sum_{i=1}^{n} \lambda_i \ln(\lambda_i).$$

We are left with a problem identical to (5.1), for which we can compute the solution explicitly:

$$\lambda_i^* := \frac{\exp(\lambda_i(G)/\gamma)}{\sum_{k=1}^{n} \exp\left(\exp(\lambda_k(G)/\gamma)\right)}.$$

We observe that the components of λ^* are ordered decreasingly. Plugging this result in our inner optimization problem, we end up with:

$$X^* := Q(G)D(\lambda^*)Q(G)^T$$

The solution X^* can be computed with a running time of $\mathcal{O}(n^3)$, that is, the order of time that the eigendecomposition of $G \in \mathcal{S}_n$ would take. As mentioned in [Nes07], the eigenvalues λ_i^* are however decreasing very rapidly, as they are inverses of exponentials. Thus, we have good chances that X^* can be accurately determined by a few of the smallest eigenvalues of G. Following this reasoning, X^* may be approximated by computing the largest eigenvalues and the corresponding eigenspaces. For a survey of different methods for computing matrix exponentials, the interested reader is referred to [ML03]. Therefore, we might compute only an approximation of $\nabla f_\mu(u)$. For an analysis of the needed precision of the approximation, we refer to [d'A08], [Bae08].

m	n	L_μ adjusted by	Alg. 2.1	opt. ratio	N_{theory}	N_{used}
10	10	1/2	0.44 sec	0.9915	1′771	890
10	10	1/4	0.31 sec	0.9916	1′252	630
10	10	1/8	0.22 sec	0.9916	886	450
10	10	1/16	0.16 sec	0.9915	626	320
10	10	1/32	0.11 sec	0.9917	443	220
10	10	1/64	0.09 sec	0.9921	313	160
10	10	1/128	0.06 sec	0.9920	222	110
10	10	1/256	0.04 sec	0.9918	157	80
10	100	1/2	22.50 sec	0.9905	3′224	800
10	100	1/4	16.92 sec	0.9921	2′280	600
10	100	1/8	11.37 sec	0.9906	1′612	400
10	100	1/16	8.50 sec	0.9922	1′140	300
10	100	1/32	5.77 sec	0.9908	806	200
10	100	1/64	5.76 sec	0.9948	570	200
10	100	1/128	2.96 sec	0.9911	403	100
10	100	1/256	2.99 sec	0.9951	285	100
100	100	1/2	119.74 sec	0.9918	3′424	1′500
100	100	1/4	88.66 sec	0.9925	2′421	1′100
100	100	1/8	64.00 sec	0.9927	1′712	800
100	100	1/16	48.51 sec	0.9935	1′211	600
100	100	1/32	33.08 sec	0.9928	856	400
100	100	1/64	25.21 sec	0.9936	606	300
100	100	1/128	17.94 sec	0.9930	428	200
100	100	1/256	17.49 sec	0.9966	303	200
1′000	100	1/2	1′310.81 sec	0.9947	3′887	2′100
1′000	100	1/4	943.73 sec	0.9950	2′749	1′500
1′000	100	1/8	708.17 sec	0.9955	1′944	1′100
1′000	100	1/16	540.14 sec	0.9954	1′375	800
1′000	100	1/32	432.88 sec	0.9963	972	600
1′000	100	1/64	308.41 sec	0.9955	688	400
1′000	100	1/128	249.19 sec	0.9963	486	300
1′000	100	1/256	195.64 sec	0.9956	344	200
10′000	100	1/2	20′260.17 sec	0.9964	4′221	2′400
10′000	100	1/4	16′257.67 sec	0.9965	2′985	1′700
10′000	100	1/16	11′871.10 sec	0.9969	1′493	900
10′000	100	1/128	8′406.94 sec	0.9965	528	300
10′000	100	1/256	8′401.32 sec	0.9982	374	300

Table A.2: *The same problem instances as in Table A.1 solved by smoothing techniques with Algorithm 2.1, but with an adjusted upper bound on the Lipschitz constant of* ∇f_μ.

B Extended numerical results

SeDuMi returns solutions that have an absolute accuracy of about 10^{-9}. As the results in Table A.1 show, the solutions computed by Algorithm 2.1 are always more accurate than what we are guaranteed by the stopping criterion. The optimality ratio, which we define as the ratio between the objective function value computed by Algorithm 2.1 and the output of SeDuMi, is always larger than the requested 0.99 due to the following reasons: First, we check every 100 iterations (for $m = n = 10$ every 10 iterations), if the stopping criterion (6.1) is satisfied. A more frequent verifying of the stopping criterion would decrease the optimality ratio. Second, and more importantly, the stopping criterion is constructed with the aid of the lower bound $\min_{X \in \Delta_n^M} \langle \sum_{j=1}^m \bar{u}_j B_j, X \rangle_F$ of the optimal value $f(u^*)$, where u^* denotes an optimal solution to the dual problem (4.3).

It is surprising that Algorithm 2.1 outperforms SeDuMi for small problem instances with matrices of dimension 10×10 and 10 constraints, provided that we sufficiently decrease L_μ without sacrificing too much the accuracy of the output (see Table A.2).

References

[AK07] S. Arora and S. Kale, *A combinatorial, primal-dual approach to semidefinite programs*, Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 2007 (D. Johnson and U. Feige, eds.), ACM, 2007, pp. 227–236.

[AS00] F. Alizadeh and S. Schmieta, *Symmetric Cones, Potential Reduction Methods and Word-by-Word Extensions*, Handbook of Semidefinite Programming (H. Wolkowicz, R. Saigal, and L. Vandenberghe, eds.), Kluwer Academic Publishers, 2000, pp. 195–223.

[Bae08] M. Baes, *Estimate sequences methods*, Tech. report, SCD/ESAT, Katholieke Universiteit van Leuven, 2008.

[BGFB94] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Studies in Applied Mathematics, SIAM, 1994.

[BV96] S. Boyd and L. Vandenberghe, *Semidefinite Programming*, SIAM Review **38** (1996), no. 1, 49–95.

[BV04] ———, *Convex Optimization*, Cambridge University Press, 2004.

[CE05] F. Chudak and V. Eleutério, *Improved approximation schemes for linear programming relaxations of combinatorial optimization problems*, Proceedings of the 11th International Conference on Integer Programming and Combinatorial Optimization, Berlin, Germany,

June 2005 (M. Jünger and V. Kaibel, eds.), Lecture Notes in Computer Science, vol. 3509, Springer, 2005, pp. 81–96.

[d'A08] A. d'Aspremont, *Smooth optimization with approximate gradient*, SIAM Journal on Optimization **19** (2008), no. 3, 1171–1183.

[Ele09] V. Eleutério, *Finding approximate solutions for large scale linear programs*, Ph.D. thesis, IFOR, ETH Zurich, 2009.

[Fan49] K. Fan, *On a theorem of Weyl concerning eigenvalues of linear transformations I*, Proceedings of the National Academy of Science of USA **35** (1949), 652–655.

[GW95] M. Goemans and D. Williamson, *Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming*, Journal of the Association for Computing Machinery **42** (1995), no. 6, 1115–1145.

[HJ96] R. Horn and Ch. Johnson, *Matrix Analysis*, Cambridge University Press, 1996.

[IPS05] G. Iyengar, D. Phillips, and C. Stein, *Approximation Algorithms for Semidefinite Packing Problems with Applications to Maxcut and Graph Coloring*, Proceedings of the 11th International Conference on Integer Programming and Combinatorial Optimization, Berlin, Germany, June 2005 (M. Jünger and V. Kaibel, eds.), Lecture Notes in Computer Science, vol. 3509, Springer, 2005, pp. 152–166.

[IPS09] ———, *Approximating semidefinite packing programs*, Tech. report, Columbia University, New York, 2009.

[Lew96] A. Lewis, *Convex analysis on the Hermitian matrices*, SIAM Journal on Optimization **6** (1996), no. 1, 164–177.

[ML03] C. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review **45** (2003), no. 1, 3–49.

[Nes05] Y. Nesterov, *Smooth minimization of non-smooth functions*, Mathematical Programming **103** (2005), no. 1, 127–152.

[Nes07] ———, *Smoothing technique and its applications in semidefinite optimization*, Mathematical Programming **110** (2007), no. 2, 245–259.

[Nes09] ———, *Primal-dual subgradient methods for convex problems*, Mathematical Programming **120** (2009), no. 1, 221–259.

[NRT] A. Nemirovski, C. Roos, and T. Terlaky, *On maximization of quadratic form over intersection of ellipsoids with common center*, Mathematical Programming volume =.

[Peñ08] J. Peña, *Nash equilibria computation via smoothing techniques*, Optima **78** (2008), 12–13.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton Mathematics Series, vol. 28, Princeton University Press, 1970.